

**Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství**

**Detekční kamerový systém automatického tísňového volání
při havárii automobilu**

**DetectionCameraSystemforAutomationEmergency Call in
Accident Case**

2014

Bc. Anna Jureková

Zadání diplomové práce

Student:

Bc. Anna Jureková

Studijní program:

N2649 Elektrotechnika

Studijní obor:

3901T009 Biomedicínské inženýrství

Téma:

**Detekční kamerový systém automatického tísňového volání
při havárii automobilu**

**Detection Camera System for Automation Emergency Call
in Accident Case**

Zásady pro vypracování:

1. Teoretický rozbor ochrany cestujících vzhledem k využití informačních monitorovacích systémů.
2. Návrh systému s detekčním kamerovým zařízením pro identifikaci počtu osob a zatížení vozidla, snímání obrazu z okolí místa nehody s časovými údaji.
3. Realizace systému s detekcí zpracování obrazu z okolí místa nehody s časovými údaji a detekcí osob ve vozidle.
4. Vizualizace a srovnání naměřených výsledků s teoretickými předpoklady.
5. Zhodnocení dosažených výsledků práce.

Seznam doporučené odborné literatury:

- [1] PHILIPS, Dwayne. *Image processing in C*. Lawrence Kansas: R & D Publications, 1994. ISBN 0-13-104548-2. 2. digitalizované vydání v roce 2000. Dostupné také z: <http://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/cips2ed.pdf>
- [2] SOBELL, Mark G. *Mistrovství v Linuxu: příkazový řádek, shell, programování*. 1. vyd. Brno: Computer Press. 2007. ISBN 978-80-251-1726-2.
- [3] DOBEŠ, Michal. *Zpracování obrazu a algoritmy v C#*. Praha : BEN - technická literatura, 2008. ISBN 978-80-7300-233-6.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Zdeněk Macháček, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě dne: 6. května 2014

.....
podpis

Poděkování

Ráda bych poděkovala doktorovi Zdeňkovi Macháčkovi za odbornou pomoc a konzultaci při vytváření této diplomové práce. Dále pak panu Jakubu J. Šimonovi za neocenitelné rady při tvorbě programu a v neposlední řadě mé rodině za pochopení a podporu v průběhu studia a psaní práce.

Abstrakt

Evropská unie vyvíjí značné úsilí na snížení dopadů automobilových nehod na zdraví lidí i majetku. Proto existují iniciativy na plošné a povinné zavedení SOS systémů na území Evropské unie. Jednou z informací, kterou je dobré zasílat tímto systémem záchranné službě, je počet osob v automobilu. Tato práce se zabývá možností rozšíření takového systému a implementací detekce osob s využitím kamerového systému. Detekce osob na základě sejmutého obrazu je jednou z možností, jak počet osob v automobilu získat. Součástí je také implementace detekce na vývojovou desku od firmy freescale i.MX53. Detekce by měla být provedena pomocí detekce obličeje na základě barvy a pohybu.

Klíčová slova

eCall, automatické tísňové volání, detekce obličeje, zpracování obrazu

Abstract

European Union put great effort to lower number of casualties and property losses due to car accidents. For this reason there are initiatives implementing mandatory SOS systems for all countries in EU. One required piece of information is about number of persons in car. This information is required by salvage and rescue service. The aim of this study is extend this system and obtain information about number of person in car by camera. This number should be detected automaticly in image. There are high requirements for accuracy and quickness. In this study the board i.MX53 from freescale is used. Programm for this board is written in C. One of the main approches in human detection is color face detection.

Keywords

eCall, automati cemergency call , face detection, image processing

Seznam použitých zkratek

Zkratka	Význam
EK	Evropská komise
EU	Evropská unie
HeERO	Harmonised eCall European Pilot
MSD	minimum set of data (minimální soubor dat)
PER	Package encoding rules
VIN	Vehicle identification number
GPS	Global positioning system
RGB	Red Green Blue
YCrCb	Y – jasový složka, Cr – červená složka, Cb – modrá složka
HSI	Hue Saturation Intensity
NCAP	New car Assessment Programme
HT	Houghova transformace
HP	Houghův prostor
PCA	Principal component analysis
FS	Face space (prostor obličejů)

Obsah

1	Úvod.....	- 3 -
2	Analýza projektu Ecall a iniciativy Evropské unie	- 5 -
	2.1 Analýza projekt HeERO	- 5 -
	2.2 Základní normativní požadavky.....	- 6 -
3	Analýza současného trhu	- 9 -
	3.1 Možnosti nabízení firmou Bosch	- 9 -
	3.2 Popis systému firmy General motors - OnStar	- 10 -
	3.3 Popis systému firmy Volvo	- 10 -
	3.4 Popis systému BMW Assist.....	- 11 -
	3.5 Analýza možností u další výrobci automobilů.....	- 11 -
	3.6 Analýza nabídky modulů pro tísňové volání u motocyklů	- 11 -
	3.6.1 Scénář při nehodě motocyklu	- 12 -
	3.6.2 Možnosti nabízené firmou Schubert.....	- 12 -
4	Zpracování obrazu – výběr metod pro implementaci	- 13 -
	4.1 Detekce hran	- 13 -
	4.1.1 Hranové detektory využívající první derivace	- 13 -
	4.1.2 Hranové detektory využívající druhé derivace.....	- 15 -
	4.2 Prahování (thresholding)	- 15 -
	4.3 Houghova Transformace	- 17 -
	4.3.1 Detekce přímky	- 17 -
	4.3.2 Detekce kružnice	- 19 -
	4.4 Detekce obličeje.....	- 20 -
	4.4.1 Detekce na základě barvy	- 21 -
	4.4.2 Analýza hlavních komponent.....	- 23 -
	4.4.3 Neuronové sítě	- 25 -
	4.4.4 Detekce obličeje pomocí Haar klasifikátorů	- 26 -
5	Návrh řešení.....	- 28 -
	5.1 Návrh jednotlivých částí detekčního řetězce	- 29 -
	5.2 Požadavky na zatížení	- 30 -
6	Řešení.....	- 31 -

6.1	Knihovna OpenCV	- 31 -
6.2	Implementace navrženého algoritmu	- 32 -
6.2.1	Převod na černobílý obraz a převod na YCrCb barevný model.	- 32 -
6.3	Detekce pohybu	- 35 -
6.3.1	Detekce barvy obličeje	- 36 -
6.4	Detekce oblastí podezřelých z přítomnosti obličeje	- 38 -
6.5	Houghova transformace	- 39 -
6.6	Upřesnění výsledku	- 40 -
6.7	Časová informace	- 41 -
6.8	Ukončení programu	- 41 -
6.9	Quick star board i.MX53	- 42 -
6.10	Implementace programu do zařízení	- 43 -
6.11	Testování	- 44 -
6.12	Časové zatížení programu	- 47 -
7	Závěr	- 48 -
8	Zdroje	- 50 -

1 Úvod

V současné době roste počet obětí automobilových nehod. Evropská Unie a výrobci automobilů se proto pokoušejí učinit jízdu autem natolik bezpečnou, nakolik je to možné. Proto se součástí výbavy automobilů staly airbagy, asistenční systémy, a nově i systémy automatického tísňového volání. Ty chce Evropská Unie zavést od roku 2015 jako povinné pro všechny nové modely aut. Dle statistik provedených Evropskou Unií by tak mělo být ročně zachráněno až 2500 životů a rovněž by měla klesnout závažnost zranění. [15]

Samy automobilky už takovéto systémy nabízejí více než 10 let. Ovšem pouze pro dražší modely aut. Z jejich zkušeností se ovšem dá čerpat při zavádění takového systému v celé EU. Jednou ze základních věcí, kterou bylo třeba rozhodnout, bylo to, která data se budou odesílat z automobilu při nehodě. Tady se do značné míry vycházelo ze zkušeností automobilek. Evropská komise stanovila tzv. minimální soubor dat (minimum set of data – MSD).

Do tohoto souboru dat patří informace o poloze, rychlosti nárazu, času nárazu, směru jízdy, typu motoru a další informace. Ovšem do povinných údajů ještě nepatří počet osob, které byly v automobilu přítomny během nehody. Především proto, že tato informace je obtížně zjištělná. V současné době je možné tuto informaci získat pomocí počtu zapnutých bezpečnostních pásů či pomocí tlakových senzorů v sedadlech. Obě tyto metody mají však svá úskalí. Například je možné, že pásy zůstanou nezapnuty.

Vzhledem k tomu, že součástí nepovinných údajů by mohla být také fotografie z místa nehody, pořízená kamerou umístěnou v automobilu, nabízí se možnost, zjistit počet osob v automobilu pomocí obrazu z kamery. Tímto tématem se bude zabývat tato diplomová práce. Zjištění počtu osob v automobilu pomocí kamerového systému se zatím neprovádí, ale mohla by to být spolehlivější metoda, než senzory z bezpečnostních pásů nebo tlakové senzory, jejichž cena není nejmenší.

Součástí této práce tedy bude navrhnout systém, který by pomocí kamery vyhodnotil, kolik osob se nachází v automobilu. Je třeba věnovat pozornost především rychlosti zpracování a přesnosti. To může být obtížné vzhledem k proměnlivým podmínkám a různým polohám osob v automobilu. Jako nejjistější se jeví detekce osob pomocí detekce obličeje. Rovněž se v práci objeví teoretický základ takového systému. Především teorie o zpracování obrazu a detekci objektů.

První kapitola této práce je věnována současným systémům pro automatické tísňové volání a projektu HeERO. Tento projekt je iniciativou Evropské unie a zabývá se plošným zavedením modulů pro automatické tísňové volání z automobilu. Součástí druhé kapitoly je průzkum trhu vzhledem k nabízeným modulům pro automatické tísňové volání. Součástí jsou možnosti nabízené jednotlivými výrobci automobilů a motocyklů. Třetí kapitola se zabývá základními metodami zpracování obrazu a možnostmi jejich praktického využití při detekci počtu osob. Na tuto část navazuje kapitola, jež rozebírá současné přístupy při detekci obličeje.

Detekce osob v automobilu by měla být provedena na základě detekce obličeje. Následující kapitoly se již zabývají návrhem, praktickým řešením a implementací programu pro detekci osob.

2 Analýza projektu eCall a iniciativy Evropské unie

Evropský parlament rozhodl, že je potřeba zvýšit bezpečnost na silnicích v EU a proto rozhodl, že se zavedou povinně moduly pro automatické tísňové volání. Tyto moduly, umístěné v automobilu, by měly přivolat pomoc v případě nehody a to i zcela automaticky, bez zásahu člověka. Tento projekt vešel ve známost pod pojmem eCall a o jeho zavedení se má postarat projekt HeERO. Evropská komise si od tohoto projektu slibuje záchranu až 2500 životů ročně a rovněž snížení závažnosti zranění.

Od října 2015 by měli výrobci nové modely automobilů povinně vybavovat tímto zařízením. Tato zařízení se již do automobilů montují více než 10 let, nicméně EK chce především dosáhnout sjednocení těchto systémů pro všechny členské státy EU. Jednotným číslem pro všechna taková tísňová volání by mělo být známé 112. Operátoři na linkách 112 by měli být vybaveni a vyškoleni tak, aby byli schopni tísňová volání přijmout. Volání by mělo být jak hlasové, pokud to bude možné, tak by měla z automobilu odejít automaticky generovaná SMS se všemi potřebnými údaji.

V současnosti se stále systém testuje, především pro navázání na tísňové linky 112. O standardizování a zavedení do provozu se pak stará projekt HeERO (Harmonised eCall European Pilot) [8] [9] [14]

2.1 Analýza projekt HeERO

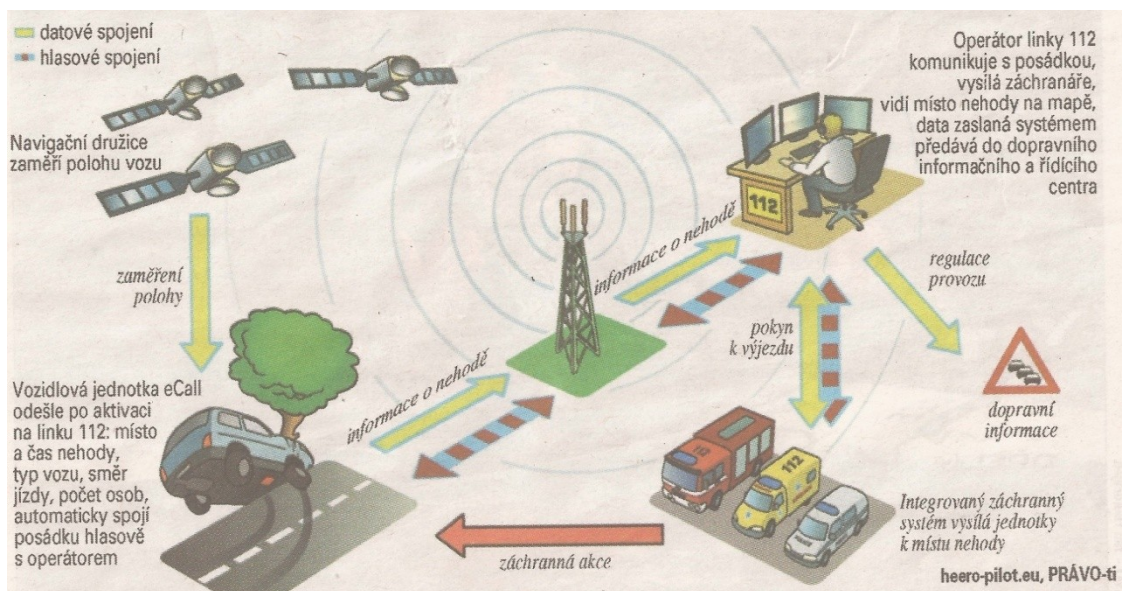
HeERO je pilotní projekt EU, který má za úkol připravit zavádění eCall 112 do reálného provozu. Hlavním úkolem projektu je především stanovit technické požadavky v souladu s evropskými standardy a normami. Otestovat stávající stav jednotlivých součástí řetězce eCall a normy, jež byly pro eCall stanoveny.

Prvním krokem je analýza stávajícího systému, jež by měl systém eCall využívat, tedy telekomunikační infrastrukturu, telefonní tísňová centra 112 a rozhraní ve vozidlech. Projekt na základě této analýzy stanoví pilotní projekty pro jednotlivé země EU, tyto projekty by měly poskytnout návod, jaké technické parametry musí být uplatněny při implementaci a na co je třeba se soustředit při testování jednotlivých částí eCall. Součástí by měla být také příručka pro pracovníky tísňových center, aby bylo jasné, jak mají s tísňovým voláním pomocí eCall nakládat.

Projekt HeERO klade důraz především na testování celého systému tak, aby na základě výsledků mohlo být sestaveno doporučení pro budoucí zavádění systému eCall. Projekt by měl probíhat v těchto zemích: Rumunsko, Německo, Česká republika, Finsko, Itálie, Nizozemí, Řecko, Chorvatsko a Švédsko. Dále pak HeERO zahrnuje testování kompatibility systému eCall s ruským systémem ERA-GLONASS.

Projekt musí také otestovat a posoudit dostupné normy a standardy pro eCall, které byly schváleny evropskými normalizačními orgány (ETSI MSG, CEN TC 278 WG 15). Následně

Výsledky by také měly stanovit požadavky na aktualizaci a modernizaci provozních a pracovních postupů, které jsou součástí záchranného řetězce. Jednotlivé země zapojené do projektu by také nové poznatky měly rozšiřovat pomocí konferencí a seminářů. [8] [9] [14]



2.2 Základní normativní požadavky

Norem pro eCall je poměrně hodně a jejich vypisování jde nad rámec této práce. Nicméně některé části by bylo dobré uvést, především co se týče požadavku na odesílání dat. Normy totiž stanovují mimo jiné i požadavky na minimální soubor dat pro odesílání (MSD). MSD jsou data, která SMS zpráva odesílaná z automobilu musí obsahovat, pokud je tedy možné je získat. Případně musí obsahovat informaci, že získání těchto informací nebylo možné. Zpráva však volitelně může obsahovat i jiná, doplňující data, z nichž některá jsou doporučena již normami. [9] [14] [15]

Pan-European eCall Operating Requirements – (PEOR)	CEN EN 16072
eCall High Level Application Protocols (HLAP)	CEN EN 16062
Third party services supported eCall – Operating requirements	CEN EN 16102
eCall Minimum Set of Data	CEN EN 15722

eCall requirements for data transmission	3GPP TS 22.101 ETSI TS 122 101
eCall Discriminator Table 10.5.135.d	3GPP TS 24.008 ETSI TS 124 008
eCall Data Transfer – General Description	3GPP TS 26.267 ETSI TS 126 267
eCall Data Transfer – ANSI-C Reference Code	3GPP TS 26.268 ETSI 126.268
eCall Data Transfer / Conformance Testing	3GPP TS 26.269 ETSI TS 126 269
eCall Data Transfer – Technical Report – Characterisation Report	3GPP TR 26.969 ETSI 126 969
Data registry procedures	ISO/EN 24978:2009

Soubor dat, která musí být zahrnuta v odesílané zprávě, jsou

- Informace o verzi MSD – počítá se s případnými změnami v budoucnu
- Informace o tom, zda bylo tísňové volání vyvoláno manuálně, jestli jde o test, je-li možno věřit udané poloze (je-li poloha přesná či nikoliv) a dále o jaký typ vozidla se jedná (motocykl, automobil atd.)
- Identifikaci vozu pomocí VIN
- Informaci o typu úložiště paliva (benzínová nádrž, nádrž na naftu, plyn atd.)
- informaci o času nehody
- informace o poloze vozidla (GPS souřadnice) a směru jízdy

Dále je pak možné přidávat další informace o předchozí poloze vozidla, o počtu pasažérů v automobilu a další. V souboru je ponechána i určitě rezerva pro možné přidání dalších informací, jako by mohla být například barva automobilu. [15]

Rovněž stojí za zmínku normy o centrech pro příjem zpráv, neboť existují dva základní modely. Jeden počítá s tzv. třetí stranou – tedy centrem, které přijme zprávu a následně informuje záchrannou službu. V podstatě jde o stejný model, který využívá většina současných výrobců automobilů, kteří nabízejí své SOS systémy. Druhá možnost je vysílání zprávy přímo na dispečink centra 112. [16]

Obě tyto možnosti mají svá pro a proti. Nicméně v obou případech musí být pracovníci schopni navázat spojení s autem a řešit situaci osob na druhé straně. Poskytnout jim pomoc a potřebné informace (uklidnit je, poskytnout rady při první pomoci atd.). V případě, že se spojení nepodaří navázat, nebo na něj nemá kdo odpovědět, na základě dat odeslaných pomocí SMS, zahájit záchranný proces – vyslání jednotlivých složek záchranné služby na místo nehody. Dispečink by také měl být schopen řešit situaci, kdy data, která přijdou SMS jsou neúplná.

Norma počítá s možností doplnění těchto dat s dostupných zdrojů jinde - například z databáze VIN kódů. [15]

Normy rovněž řeší i požadavky na zařízení umístěné v automobilu. Za zmínku například stojí požadavek na hlasové spojení – které by mělo být vždy možno navázat. Jedním z hlavních požadavků je také schopnost bezpečně odesílat data. Norma stanovuje, že alespoň v 95% případů, musí být odesílání úspěšné a musí trvat pod 20sekund a to v místech kde je mobilní síť silnější než 99dBm, přičemž v daných 20 sekundách je možné vysílání opakovat.

Normy rovněž stanovují požadavky na kódování dat na způsoby přenosu dat, na jejich rychlost, přesnost a dostupnost na odlehlých místech. Data by měla být prezentována v abstraktní syntaktickém zápisu, ASN.1 Packedencodingrules (PER unaligned). V příslušných normách jsou všechny tyto požadavky přesně stanoveny a rozepsány. [17]

3 Analýza současného trhu

V současnosti se již prodávají automobily s moduly pro automatické tísňové volání, nicméně tato zařízení ještě nejsou v základní výbavě, jak by tomu mělo být v EU od roku 2015. Téměř všichni výrobci automobilů nabízejí takováto zařízení, ačkoliv ne u všech svých modelů.

Poprvé s takovýmto zařízením přišla na trh firma BMW, která nabízí toto zařízení jako nadstandardní výbavu do dražších aut již od roku 1999. Firma si sama řídila operační centrum pro příjem takovýchto volání. Brzy následovaly i další výrobci automobilů – Volvo (2000), Mercedes – Benz (2001), Peugeot, General Motors, Citroën a další.

Dnes již téměř všichni výrobci automobilů vyrábějí, a do svých aut montují, podobná zařízení. Taktéž mnohdy provozují svá vlastní centra příjmu tísňových volání, která fungují navíc také jako asistenční centra pro případ poruchy, zdravotních potíží nesouvisejících s nehodou, vyhledáváním cesty, dopravními informacemi a dalšími doplňkovými službami. Výrobci automobilů sami vyhodnotili, že tím získají více zákazníků a služeb bude více využíváno. Za zmínku ovšem stojí, že možnost využití této služby je většinou placené, v mnoha případech se předplácí možnost celoročního využívání. Samozřejmostí je ovšem možnost využívání služeb bezplatně po nějakou dobu při koupi nového automobilu, či její zrušení v případě, že o takovou službu nemáme zájem. Díky tomu pak není služba prodělečná. Co se s těmito službami stane po zavedení panevropského tísňového centra je otázka do budoucna.

Výrobci automobilů ovšem tuto službu většinou nabízejí pouze lokálně, většinou v jednom státě. Celoevropský či přímo celosvětový provoz by byl prodělečný. Výrobci automobilů většinou spolupracují v daném státě s některým lokálním operátorem například BMW v Německu s Vodafone. Iniciativa Evropské unie počítá s využitím linky 112. [20] [23]

3.1 Možnosti nabízené firmou Bosch

Firma Bosch se v posledních letech silně angažuje v nově vznikajícím systému eCall. Jednak vyrábí zařízení pro automatická tísňová volání, které pak prodává výrobcům automobilů, jako je BMW nebo Citroën. Dále pak již provozuje mezinárodní evropské centrum pro příjem tísňových volání, z takovýchto zařízení. Jako mezinárodní centrum, společné pro celou Evropu, má dle firmy Bosch jejich řešení tu výhodu, že jsou schopni navázat spojení s účastníky nehody i místními složkami záchranné služby v jejich vlastním jazyce. Pro příklad, pokud by se německy mluvící řidič naboural ve Španělsku, centrum Bosch s ním naváže spojení v němčině a zjistí od něj podrobnosti o nehodě. Následně kontaktuje španělskou záchrannou službu a všechny potřebné informace jim poskytne ve španělštině.

Firma Bosch začala s budováním a provozem těchto center již v roce 2004. Proto má již mnohaleté zkušenosti s příjmem tísňových volání. V roce 2012 byla tato služba přístupná již v 9 zemích a to v Německu, Francii, Itálii, Španělsku, Spojeném Království, Rakousku, Nizozemí, Belgii a Švýcarsku. [23]

3.2 Popis systému firmy General Motors - OnStar

OnStar je dceřiná společnost firmy General Motors, která se zabývá výrobou automobilů, automobilových součástí a vybavením. Firma OnStar se zaměřuje na vybavení automobilů jako je turn-by-turn navigace, hands free komunikace, bezpečnostní systémy a další.

OnStar operuje zejména v severní Americe (Kanada a USA) a v Číně. Mimo dalších služeb nabízí i možnost příjmu automatického tísňového volání. V případě, že dojde k nehodě, může automobil vybavený systémem OnStar vyslat automatické tísňové volání k centru, které jede v provozu 24/7. S operátorem tohoto centra naváže zařízení z automobilu hlasové spojení. Operátor se pak následně zeptá, zda řidič potřebuje pomoc. Je-li pomoc vyžadována nebo nedojde-li k navázání hlasového spojení s účastníky nehody, jsou na místo nehody vyslány jednotlivé složky záchranné služby. Těm může operátor rovněž sdělit informace, které zaslalo zařízení z automobilu. Sem patří informace o mnohočetných nárazech, informace z rollover senzorů, rychlost automobilu při nárazu, zda došlo k vystřelení airbagů a samozřejmě GPS souřadnice.

Na rozdíl od mnoha jiných podobných zařízení, systém od firmy OnStar vysílá tísňová volání nezávisle na tom, zda došlo, či nedošlo k vystřelení airbagů. Senzory jsou umístěny na několika místech v automobilu tak, aby byly schopné zaznamenat nárazy z různých směrů a úhlů.

Systémy od OnStar jsou, stejně jako ostatní podobné systémy, vybaveny i tlačítkem pro manuální spojení s operačním centrem. Posádka automobilu tak může rychle vyžádat asistenci i v dalších krizových situacích. Jako příklad je možno uvést náhlé zdravotní potíže řidiče (např. srdeční příhoda) nebo pokud jsme svědky dopravní nehody, nikoliv však účastníky. Případně můžeme přivolat pomoc při jakýchkoliv dalších potížích nás nebo osob v okolí (porouchaný automobil, ztracené dítě atd.)[19]

Systémy OnStar je možné najít nejen v automobilech General Motors, ale také u jiných automobilů. Příkladem může být Lexus firmy Toyota. [19]

3.3 Popis systému firmy Volvo

Volvo uvedlo svůj systém na trh v roce 2000. V té době mělo vedoucí pozici při zavádění elektronických systémů do automobilů. Systém měl fungovat dle standardního protokolu, který známe dnes. Po vystřelení airbagů je informována jednotka On Call Volvo, která je zároveň napojena na GPS a vestavěný mobilní telefon. Následně je odeslána textová zpráva do alarm centra a je navázáno hlasové spojení, případně zavolána hlasová služba. Systém byl nejprve zaveden ve Švédsku a ve Spojených státech. V Evropě Volvo využívalo sítě GSM, v severní Americe i zpráv přes satelit, neboť se zde nacházejí místa bez pokrytí sítí GSM. V severní Americe však bylo Volvo vytlačeno firmou OnStar (GM). [26]

3.4 Popis systému BMW Assist

Firma BMW pracuje na Evropském trhu nejdéle. Od roku 2007 provozuje v Olchingu u Mnichova callcentrum, kde jsou směřována tísňová volání z aut BMW v Německu. Callcentrum má okolo 50 psychologicky vyškolených pracovníků, kteří mají za úkol vyhodnotit závažnost nehody, posbírat všechny potřebné informace a v případě nutnosti zavolat záchrannou službu. Té pak předají všechny relevantní informace, aby mohla být záchranná akce co nejrychlejší a nejefektivnější.

V první třetině roku 2011 řešili operátoři tohoto centra na 1600 volání. Je však třeba zmínit, že některá byla aktivována stisknutím tísňového tlačítka. Takto také jeden zákazník přivolal policii, když uviděl v samoobsluze zloděje. Je tedy dobré zmínit, že BMW Assist nenabízí pouze služby pro volání při nehodě, ale i další služby v případě potřeby. Například při poruše automobilu či při navigaci.

BMW za svůj počín obdrželo mezinárodní ocenění od komisařů z Euro NCAP. [27]

3.5 Analýza možností u dalších výrobců automobilů

Toyota nabízí ve svých automobilech více druhů systémů, které obsahují jako jednu z funkcí automatické tísňové volání v případě nehody.

Za zmínku stojí především SafetyConnect který je dostupný u modelů Toyoty od podzimu 2009. Řidič, který se rozhodne využívat služeb SafetyConnect může vyžadovat pomoc při několika situacích – při krádeži automobilu, v případě poruchy, nehody či při jiné pohotovostní situaci. Služba je ovšem dobrovolná, proto pokud ji majitel automobilu nehodlá využívat, může ji zrušit.

Citroën nabízí službu eTouch. Služba funguje na podobném principu, jako u ostatních automobilek. Citroën svou službu poprvé nabídl zákazníkům v roce 2003. Od roku 2010 mohou službu využívat i majitelé Citroënů s tímto zařízením v České republice.

Firma Ford vybavuje své vozy systémy pro přehrávání hudby, ovládání vozu, automatickou diagnostiku a další. Součástí těchto systémů je také aplikace 911 Assist. Ten je schopen navázat spojení s centrem záchranné služby v případě nehody. Pokud dojde k vystřelení airbagů, celý systém 10 sekund počká, zda se posádka automobilu nerozhodne tísňové volání zrušit. Pokud k manuálnímu zrušení během 10 sekund nedojde, zařízení před připojením telefonu naváže zvukové spojení se záchrannou službou. V případě, že nikdy v automobilu není schopen odpovědět na dotazy, systém sám podá potřebné informace, jako je místo nehody čas nehody atd. [20] [21] [22]

3.6 Analýza nabídky modulů pro tísňové volání u motocyklů

Firma Schuberth, která se zabývá výrobou přileb pro motocyklisty, v roce 2013 představila zařízení pro automatické tísňové volání pro motocykly.

Součástí zařízení je několik senzorů, které jsou umístěny jak na motorce, tak na přilbě jezdce. Jednotka, která se připojuje k motorce, vyhodnocuje, zda došlo k nehodě, na základě senzorů náklonu. Pokud sklon motorky padne pod jistou úroveň, systém vyhodnotí, že došlo k nehodě.

Dále je v jednotce pro vyhodnocování nehody, také zařazen akcelerometr, který vyhodnocuje nehodu na základě negativního zrychlení. Pokud je hodnota netypická, velké nárazové zpomalení, systém opět vyhodnotí, že došlo k nehodě a vyšle tísňové volání

Další součástí jsou senzory umístěné přímo na přilbě jezdce. Jeden z nich zjišťuje, zda neselhalo spojení mezi zařízením na helmě a na motorce. Spojení selže, pokud se jezdec vzdálí od motorky. Dále je součástí také senzor, který kontroluje, zda je modul připojen na baterii motorky, pokud tomu tak není, začne alarmovat. Modul má samozřejmě vestavěnou baterii, která zařízení napájí v případě, že k odpojení opravdu dojde.

Dalším senzorem umístěným v helmě je akcelerometr, stejně jako tomu je u motorky. Ten zjišťuje, zda helma neutrpěla prudký náraz. Nakonec je také třeba zmínit, že řidič může sám vyslat tísňové volání manuálně, pomocí tlačítka, stejně jako je tomu v osobních automobilech. Stejně jako obdobná jednotka v automobilu má i zařízení umístěné na motocyklu vestavěný GPS přijímač, integrovanou mobilní SIM kartu a mikrofon, s dosahem okolo 3 metrů. [28]

3.6.1 Scénář při nehodě motocyklu

V případě, že jeden, nebo více senzorů vyhodnotí, že došlo k nehodě, je vyslána textová zpráva do centra, které zabezpečuje příjem tísňových volání. Zpráva obsahuje údaje o poloze získané z GPS přijímače a data o osobě, která vyžaduje pomoc (tato data jsou uložena v zařízení). Rovněž je v takovém případě navázáno zvukové spojení motocyklu s centrem přijímajícím tísňové volání. Pracovník centra se pak pokusí navázat spojení s řidičem, pokud řidič vyžaduje asistenci nebo spojení s ním není navázáno, dojde k uvědomění nejbližšího centra záchranné služby a na GPS souřadnice získané z textové zprávy, je vyslána záchranná služba. [28]

3.6.2 Možnosti nabízené firmou Schubert

Zařízení firmy Schubert je možné na internetu zaregistrovat a využít dodatečných možností od jejich partnerů. Například partnerské firmě Björn Steiger Stiftung můžete umožnit založení osobní pohotovostní složky. V té by měla být uložena data, jako je krevní skupina, alergie na léky či prodělané nemoci a operace. Ty mohou být odeslány spolu s GPS souřadnicemi a dalšími informacemi záchranné službě, která na jejich základě, může poskytnout profesionální ošetření. [28]

4 Zpracování obrazu – výběr metod pro implementaci

Zpracování obrazu je disciplína informatiky, která se dynamicky v současné době rozvíjí. Jedním z důvodů je to, že obrazová informace je jedním ze základních komunikačních a informačních kanálů. I při detekci počtu osob v automobilu bude třeba využít některých metod zpracování obrazu. Především pak pro detekci obličejů. Detekce obličejů je sama o sobě komplexním problémem a možnosti detekce, jež by mohly být využity při implementaci, budou řešeny v další kapitole. Součástí práce by mělo být i využití Houghovy transformace k detekci očí v obličejích. Tato metoda by měla být zkombinována s některou z dalších metod detekce, aby detekce byla co nejpřesnější a nejrychlejší.

Houghova transformace je metoda pracující s parametrickými rovnicemi zkoumaných útvarů. Pro její využití je tedy třeba znát parametrickou rovnici struktury. Proto se tato metoda používá převážně k detekci kružnic, přímek či elips. Houghova transformace však také vyžaduje předzpracování obrazu a to především převedení na šedotónový obraz a následně detekování hran. [2] [4]

4.1 Detekce hran

Hrany jsou místa obrazu, které nesou největší informaci. Proto je někdy dobré je detekovat a oddělit je od zbytku obrazu. Hrany se nacházejí tam, kde se prudce mění jas. Toho se využívá při jejich detekci. Výstupní obraz po detekci hran by měl obsahovat pouze informaci o tom, kde se hrany nacházejí a všechny ostatní informace (například barva), by měly být zanedbány. Detekci hran je dobré věnovat pozornost, neboť některé metody, například Houghovy transformace, závisí na této metodě zpracování obrazu.

Jak už bylo řečeno výše, hrana je místo kde se skokově mění jas. Hrana tedy, oproti homogenním plochám, obsahuje vyšší frekvence. Obecně platí, že čím je větší změna jasu, tím je hrana ostřejší. A také platí, že ostřejší hrany je možné lépe detekovat.

K detekci se v praxi používají diferenciální operátory prvního a druhého řádu. Gradientní metody neboli metody prvního řádu využívají maxima první derivace. Neboť čím ostřejší je hrana, tím větší bude lokální maximum v obrazu. U operátorů druhého řádu se pak pracuje s druhou derivací, kde se hledá průchod nulou. [2] [4] [12]

4.1.1 Hranové detektory využívající první derivace

Tyto detektory se také nazývají gradientními detektory a aproximují nejčastěji absolutní hodnotu a úhel gradientu jasu. Absolutní hodnota gradientu je nezávislá na směru, úhel pak na směru závisí. Pro výpočet se využívají kompasové masky, jež aproximují parciální derivace pro výpočet absolutní hodnoty gradientu i úhlu. Tato metoda je však citlivá na kvalitu obrazového snímku, a pokud je snímek zašuměný, podává špatné výsledky, neboť tyto masky zkoumají diferencii pouze dvou bodů. Proto se často užívá masek, jež počítají diferencii více prvků. [2]

Běžně se při detekci hran pomocí první derivace využívají konvoluční masky, kam patří například Sobelův operátor, Kirschův operátor či operátor Prewittové. Jejich příklady je možné vidět níže.

$$\begin{array}{ll}
 \text{Prewittové :} & \begin{vmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix} & \text{Sobelův :} & \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix} \\
 \text{Kirschův :} & \begin{vmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 3 \end{vmatrix} & \text{Robinsonův :} & \begin{vmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{vmatrix}
 \end{array}$$

Tyto masky však detekují hrany pouze v jednom směru. Proto se často užívá jedné masky tak, aby pracovala ve všech směrech. Matice se pootočí a pro každý směr pak získáme jinou matici, kterou použijeme na zkoumaný obraz. Příkladem jsou následující masky pro detekci hran Sobelova filtru.

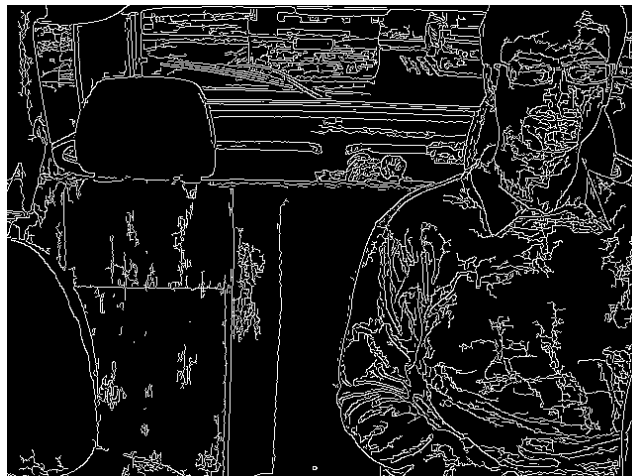
Sobel:

$$\begin{array}{cccc}
 \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix} & \begin{vmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{vmatrix} & \begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix} & \begin{vmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{vmatrix} \\
 \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix} & \begin{vmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{vmatrix} & \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} & \begin{vmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{vmatrix}
 \end{array}$$

Konvoluční masky se na obraz užívají následovně. Pro každou hodnotu jasu v obraze je vypočtena nová hodnota. Nejprve je matice přiložena na první element obrazu a následně je provedena konvoluce (hodnoty ležící pod sebou se pronásobí a pak je z nich vypočten součet). Po získání jedné hodnoty se přistoupí k dalšímu pixelu, přičemž je třeba vypočítat novou hodnotu pro každý pixel v obraze. Touto metodou se zvýrazní hrany. Tam, kde se hodnoty příliš nemění, se výsledek konvoluce bude rovnat hodnotám blízko nuly.

V zásadě nezáleží na tom, jakým způsobem bude maska obraz propočítávat. Zda z pravého horního rohu, či z pravého dolního. Je však třeba mít na paměti, že některé detektory jsou pouze jedno směrové. Detekují tedy hranu pouze z jednoho směru. Proto se mnohdy užívá 8 masek a tím je možné detekovat hrany z 8 základních směrů. Rovněž je možné použít jednu masku a aplikovat ji na obraz z různých stran.

Za zmínku stojí ještě Cannyho detektor, který se od výše zmíněných odlišuje. Neboť prvním krokem je zde užita konvoluční matice, která má vyhladit celý obraz a tím také potlačit šum. Až poté se aplikuje některý výše zmíněný operátor. Pak jsou vypočteny absolutní hodnoty a úhly a následně jsou ještě ztenčeny zjištěné hrany. Nakonec je obraz prahován s hysterezí. Příklad je možné vidět na obrázku 4.1.[2] [12]



Obrázek 4.1: detekce hran pomocí Cannyho detektoru

4.1.2 Hranové detektory využívající druhé derivace

Zde se stejně jako u první derivace využívá masky, aby se aproximovala druhá derivace. Příkladem takové masky je

$$\begin{vmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{vmatrix}$$

Tato maska aproximuje diferenciální rovnici druhého řádu. Opět je možné masku pootočit a vypočítávat tak hodnoty pro více směrů. Přičemž tato maska detekuje ve směru šikmo. Tato metoda je však vysoce citlivá na šum. To je možné potlačit užitím průměrovací matice, kdy je obraz před detekcí hran vyhlazen. Oba tyto procesy je možné také propojit do jedné matice. Příklad je vidět níže.

$$\begin{vmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{vmatrix}$$

Po užití takovéto matice je ještě třeba detekovat průchody nulou a získat tak binární obraz. Tedy obraz kde se budou vyskytovat pouze bílé hrany a černé pozadí.

4.2 Prahování (tresholding)

Prahování se řadí do základních metod pro segmentaci obrazu. Existuje několik základních metod pro prahování. V základě vychází tyto metody především z histogramů a převodních charakteristiky. Rovněž je možné prahovat na základě statistik, či případně metodou pokus-omyl. Posledně zmíněná metoda není příliš praktická, nicméně při úpravě malého objemu dat je možné s ní dosáhnout dobrého výsledku.

Nejjednodušší bude si základní princip prahování vysvětlit na bimodálním histogramu. Takový histogram bychom v ideálním případě získali z šedotónového obrazu, kde jsou světlé objekty v popředí a za nimi je pak tmavé pozadí. V histogramu nám pak vzniknou dvě oddělená maxima. Obvyklý postup pro oddělení objektů je zvolení prahové hodnoty, dle které se uvedené dva módy oddělí. V zásadě pak platí pro bod (x, y) a prahovou hodnotu T . [2] [4]

$$f(x, y) > T \text{ bod náleží objektu} \quad (4.1)$$

$$f(x, y) < T \text{ bod náleží pozadí} \quad (4.2)$$

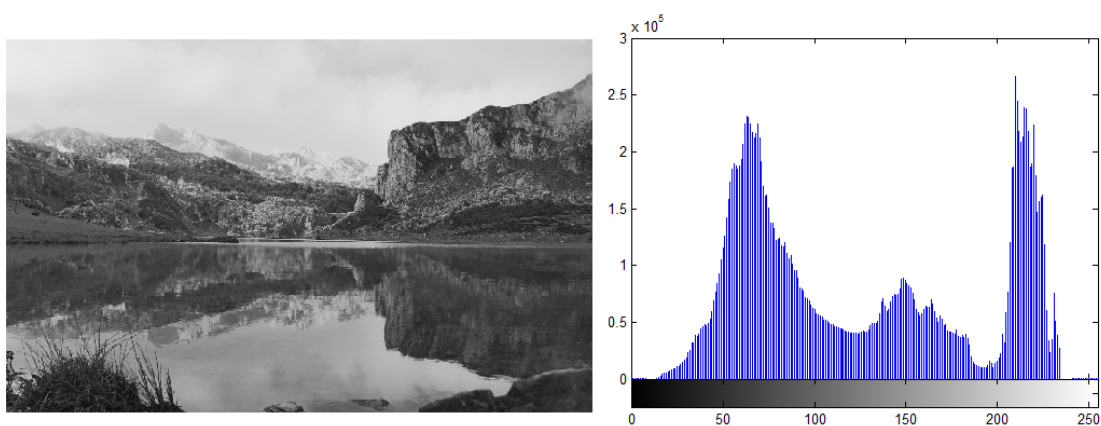
Ovšem na místo dvou maxim můžeme mít v histogramu maxim i více. Potom musíme použít víceúrovňové prahování, kdy body jednotlivým objektům přisuzujeme podle

$$T_1 < f(x, y) \leq T_2 \text{ Bod náleží prvnímu objektu} \quad (4.3)$$

$$f(x, y) > T_2 \text{ Bod náleží druhému objektu} \quad (4.4)$$

$$f(x, y) < T_1 \text{ Bod náleží pozadí} \quad (4.5)$$

Obdobně bychom postupovali i v případě, že bychom v histogramu získali další maxima. Vždy je však třeba mít na paměti co chceme ještě ponechat jako pozadí a co nikoliv. V reálném histogramu nemusí být jednotlivá maxima výrazně oddělena obrázek č. 4.2. Proto může být obtížné od sebe jednotlivé části obrazu oddělit. [2]



Obrázek 4.2: - obecný snímek s histogramem

Prahování je tedy v podstatě testování funkce $T = T[x, y, p(x, y), f(x, y)]$ vůči původní funkci $f(x, y)$ v níž jsou reprezentována úroveň šedi daného obrazového bodu.

Prahovaný výstupní obraz je definován jako

$$g(x, y) = \begin{cases} 1 & \text{pokud } f(x, y) > T \\ 0 & \text{pokud } f(x, y) < T \end{cases} \quad (4.6)$$

Pak tedy ve výsledném obrazu mají body objektu přiřazenu hodnotu 1 (běžně hodnotu 255) a body pozadí hodnotu 0. V praxi se nejčastěji používají osmibitové vyjádření obrazu. Kde hodnota 0 zastupuje černou a hodnota 255 bílou. Výsledný obraz po prahování je potom pouze dvouúrovňový, kde černá zastupuje pozadí a bílá objekty.

Pokud práh T závisí pouze na funkci $f(x, y)$ jde o prahování globální. Takovéto prahování si však můžeme dovolit použít pouze tam, kde můžeme vhodně ovlivňovat jas původního obrazu. V reálu se však může stát, že takový obraz nemáme. Další možností je pak prahování lokální kdy T nezávisí pouze na $f(x, y)$, ale také na $p(x, y)$, kde $p(x, y)$ představuje lokální vlastnosti obrazového bodu. O krok dál je pak ještě prahování dynamické kde T mimo výše zmíněného, závisí také na souřadnicích x, y . [2] [4]

4.3 Houghova Transformace

Jde o metodu, jež je schopná detekovat v obraze struktury, u nichž je znám jejich analytický popis. Jako dobrý příklad může být detekování přímk, kružnic či elips. Metoda byla poprvé popsána Paulem Houghem a podle něj je také pojmenována. Dále pak byla rozšiřována dalšími vědci.

Abychom mohli Houghovu transformaci použít, je obraz třeba nejprve předzpracovat. Houghova transformace totiž vypočítává všechny možné existence dané struktury pro každý bod v obrazu a následně je ukládá do tzv. Houghova prostoru. Na základě maxima, případně prahu, se pak dá rozhodnout, kde leží hledané struktury. Toto se však nedá použít, pokud bychom měli vypočítávat hodnoty Houghova prostoru pro každý bod v reálném obrazu. Předně by toto bylo extrémně náročné na čas i paměť, a navíc bychom nebyli schopni zjistit požadovanou informaci. Proto se jako vstup pro HT používá binárního obrazu, který získáme po detekci hran a prahování. Výstupem je pak Houghův prostor, který obsahuje informací kandidátů na hledanou strukturu. Výstupní parametrický prostor může být n -rozměrný, v závislosti na tom, kolik neznámých máme v parametrické rovnici struktury. Pro přímku bude prostor dvourozměrný, pro kružnici pak trojrozměrný. [2]

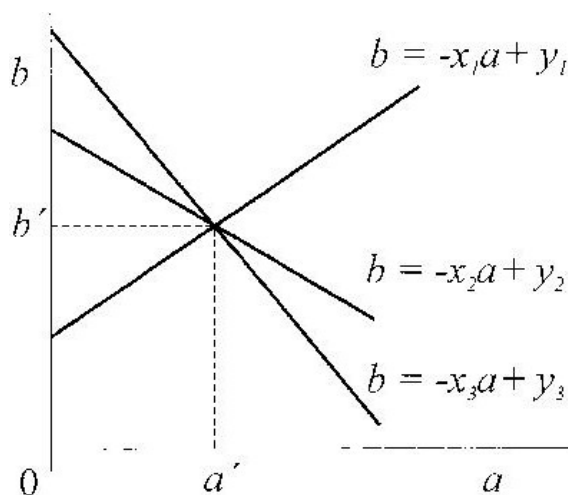
Z výše uvedeného je ale jasné, že metoda může být značně početně, paměťově a časově náročná. Proto je nutné ji maximálně optimalizovat. Zároveň je však metoda výhodná v tom, že je poměrně robustní a je tak schopná detekovat struktury i v nekvalitním snímku. Přesnost a rychlost zpracování však také závisí na kvalitní detekci hran, což je při implementaci třeba mít na paměti.

4.3.1 Detekce přímk

Za nejjednodušší tvar, který se pomocí HT detekuje je přímk (případně úsečka). Přímku můžeme vyjádřit její parametrickou rovnicí.

$$y = ax + b \quad (4.7)$$

Kde a a b jsou parametry přímky, a je směrnice dané přímky a b je úsek, jež přímka vytíná na ose y . Je dobré si uvědomit, že každé přímce odpovídá jiná dvojice parametrů a , b . Je tedy možné si představit rovinu, ve které na jednu osu vyneseme a na druhou b . Jeden bod v takovéto rovině pak bude označovat jednu konkrétní přímku. Takto vzniklý prostor se občas nazývá Houghův prostor obrázek 4.3. [2]



Obrázek 4.3: Příklad Houghova prostoru pro přímku [12]

Pokud tedy vytvoříme všechny možné přímky pro všechny body v obraze (myslíme obraz po prahování, kde hodnoty jsou pouze 1 nebo 0) v místě kde se protne nejvíce přímek, tedy kde bude „četnost bodu“ $[a, b]$ největší jsou parametry dané přímky.

Tato metoda však není nejvhodnější například pro přímky kolmé, kde a se může blížit nekonečnu. Proto se v praxi používá trochu jiného přístupu a to je nahrazení parametrické rovnice rovnicí v normálovém tvaru tedy

$$r = x \cos \varphi + y \sin \varphi \quad (4.8)$$

Opět platí, že pro každý bod v obraze teoreticky existuje nekonečné množství rovnic, jež pro něj mohou odpovídat, ale pro každou přímku odpovídá pouze jedna rovnice. Vytvoření souřadného systému s ρ a θ – jednomu bodu bude odpovídat nekonečné množství dvojic. Pokud budeme měnit pro daný bod θ pak ρ se bude měnit tak, že vznikne sinusoida.

Stejný postup je možné aplikovat na jakýkoliv tvar, který jsme schopni analyticky popsat. Daná struktura pak bude mít parametry v místě, kde se protne maximum sinusoid. Pokud chceme najít pouze jednu přímku v obraze, stačí nám toto maximum, pokud chceme detekovat přímek více, musíme stanovit práh, jenž bude odpovídat počtu průniků. Tento práh pak bude určovat, co se ještě bude považovat za přímku a co nikoliv.

4.3.2 Detekce kružnice

Další strukturou, kterou je možné takto detekovat je kružnice. Nalezení je pouze obdobou při hledání přímky. I zde potřebujeme rovnici kružnice, středová rovnice má tvar

$$r^2 = (x - a)^2 + (y - b)^2 \quad (4.9)$$

Kde r je poloměr kružnice a a, b jsou souřadnice, na nichž leží střed. Vzhledem k tomu, že jsou zde 3 proměnné, bude parametrický Houghův prostor (HP) trojrozměrný. Je možné si to představit tak, že na ose x bude ležet parametr a , na ose y parametr b a na ose z parametr r . Stejně jako přímku je možné i kružnici zapsat pomocí parametrické rovnice, která má tvar

$$x = a + r \cos \varphi \quad (4.10)$$

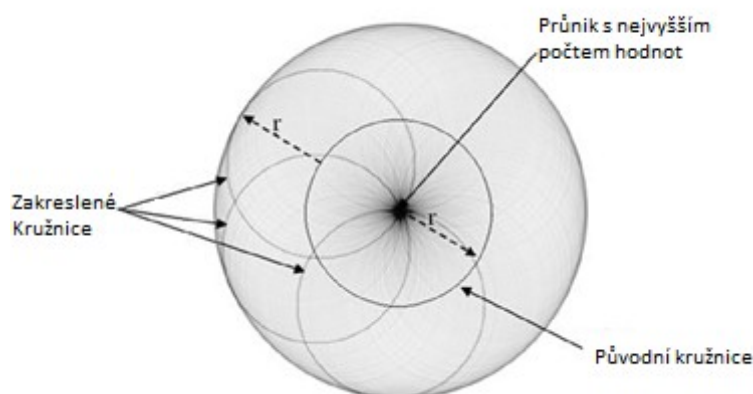
$$y = b + r \sin \varphi \quad (4.11)$$

kde φ leží v intervalu $0 - 2\pi$. Pokud za x a y budeme dosazovat data ze zkoumaného obrazu (opět z obrazu po detekci hran). Budeme schopni následně dopočítávat všechny možné a, b a r pro dané body. To by však bylo značně náročné, většinou však nehledáme kružnice o jakémkoliv poloměru, ale o poloměru, který je více či méně znám. Výpočet tak můžeme značně zkrátit. Pro zjednodušení je uveden výpočet pro případ, kdy je hledaný poloměr jasně daný. Tím se získá pouze dvourozměrný prostor. Z předchozích rovnic je možné vyjádřit a, b tedy

$$a = x - r \cos \varphi \quad (3.12)$$

$$b = y - r \sin \varphi \quad (3.13)$$

Souřadnice z binárního obrazu se poté dosadí do těchto rovnic a pomocí φ jsou vypočteny kružnice, které se promítnou do parametrického prostoru, kde se protnou v určitém bodě. Tím se získá poloha hledané kružnice. Pro názornost je zde uveden obrázek 4.4. V praxi však není poloměr kružnice přesně znám, je pak tedy možné použít určité rozmezí hodnot pro poloměr r . Vypočítávat všechny možné existující kruhy, byť teoreticky jen do poloměru v daném obraze, by bylo značně náročné a nepraktické. Díky tomu je však následně získán parametrický prostor trojrozměrný. Dosazením do rovnic 3.12 a 3.13 je získána množina všech možných řešení pro a, b, r . V parametrickém prostoru pak půjde o kužel. V případě, že jsou dosazeny všechny body ležící v binárním obraze, výsledkem je množina kuželů, jež se protnou v určitém, případně určitých bodech, které budou odpovídat parametrům kružnic. Na daných souřadnicích tedy pak budou ležet kružnice.



Obrázek 4.4: Houghův prostor pro detekci kružnic[25]

Algoritmus pro výpočet by prakticky byl následující. Vytvoří se akumulátor pomocí trojrozměrného parametrického prostoru. Pro každý bod vstupního binárního obraz se vypočte odpovídající a , b a následně r pro určené rozmezí poloměrů a rovněž se dosadí φ . Přičemž hodnoty r a φ se dosazují po předem určeném kroku (není možné je dosazovat spojitě). Na všech vypočtených pozicích pak dojde k inkrementaci hodnot. Takto získáme HP obdobný tomu, který se získá při hledání přímk. V místě, kde se protne nejvíce kružnic, pak leží body odpovídající parametrům kružnice v reálném obrazu. [2]

4.4 Detekce obličeje

Obecně jsou metody pro detekci obličeje založené na tom, že chceme získat segment obrazu, jenž obličeji odpovídá. Při detekci obličeje se setkáváme s několika problémy, jež nastávají v reálných situacích. Jsou to

- tvary v obličeji, jež obličeji neodpovídají – různá znetvoření, případně prvky, jež se v obličeji nevyskytují (piercing, brýle),
- natočení obličeje – je-li obličej snímán ze strany, záklon hlavy atd.
- pózy obličeje jako je úsměv nebo škleb
- změny jasy při různém osvětlení
- zakrytí obličeje – šátkem, šálou, kuklou.

Díky těmto nevýhodám je někdy těžké obličej detekovat. Proto se také nedá očekávat, že metody pro detekci obličeje budou fungovat bezchybně. Nicméně mnohé dnešní algoritmy pro detekci obličeje jsou schopny dosáhnout solidních výsledků okolo 96% úspěšnosti. Přičemž za neúspěch se považuje detekce oblasti, jež není obličejem, stejně, jako obličej, který nebyl detekován.

Rozpoznávání obličejů má mnohostranné využití a v zásadě se dá rozdělit na 3 základní podskupiny a to na identifikaci obličeje, jeho klasifikaci a rozpoznávání pohlaví. Jednou z užitečných aplikací je prohledávání davu, osobní identifikace, bezpečnostní vstupy a další. [7]

Existuje několik základních obecných druhů detekce.

Znalostní metody, které jsou založeny na základě obecných znalostí lidského obličeje. Tím se myslí její základní komponenty, jako jsou oči, ústa, nos a jejich vzájemný vztah. Rysy v obličeji se svou vzájemnou polohou a velikostí. Podobné poznatky se rovněž využívají v umění při kresbě tváře při rozvržení kresby. Tyto metody pro svou funkci vyžadují, aby zkoumané struktury byly nejprve v obličeji detekovány. Po jejich nalezení je pak možné zkoumat jejich polohu a vzájemné vztahy. Správná a přesná lokalizace rysů však může být poměrně náročná. Stejně jako zakódování znalostních pravidel, podle nichž by se pak mělo vyhodnotit, zda jde, či nejde o obličej. Takovéto metody jsou velice citlivé na změnu polohy obličeje i intenzitu osvětlení, a pokud nejsou správně navrženy, nepodávají dobré výsledky.

Další skupinou jsou metody založené na neproměnlivosti lidských rysů. Základem těchto metod je předpoklad, že v obličeji existují příznaky, které jsou neproměnlivé při změně polohy či osvětlení. Tedy, že je možné je detekovat v jakékoliv poloze obličeje. Zde se mohou uplatnit rysy, jako jsou oči, ústa a nos, případně barva obličeje, která je také neproměnlivá. Při extrakci příznaků jako jsou oči, ústa a nos se nejčastěji využívá hranových detektorů. Detekci barvy je možné provádět na základě více barevných modelů. V neposlední řadě do této kategorie spadá také detekce na základě textury. Zde se pak pracuje s tím, že textura obličeje je v zásadě odlišná od okolí. Počítá se s tím, že oblasti, kde se nacházejí ústa, oči a nos, jsou oblasti tmavší než zbytek tváře.

Dále je možné zkoumaný obraz srovnávat s referenční šablonou. Přičemž šablona je dopředu vytvořena buďto ručně nebo pomocí naprogramovaných funkcí. Zde je nutné nejprve korelovat obraz a šablonu. Následně je přítomnost obličeje vyhodnocena na základě korelační funkce. Problém s tímto přístupem je v tom, že obličej v zásadě nemají ve zkoumaném obraze stálou velikost. Je tedy potom potřeba měnit i velikost šablony. Rovněž je nutné mít více šablon pro různé náklony obličeje. Tím pak roste početní a časová náročnost detekce.

Podobný přístup je také u metod detekujících obličej na základě vzhledu. Zde však nejsou porovnávány obrazy se šablonou, ale jsou vytvořena pravidla pro rozhodování na základě strojového učení na trénovacích množinách. Nejprve jsou vytvořeny distribuční modely na základě předložených příkladů s obličejí a bez nich a ty jsou pak využity při detekci. Metod pracujících na tomto principu je mnoho, ale bývají poměrně náročné na implementaci a značně závisí na vhodné volbě trénovacích množin.

Toto je jen hrubé základní rozčlenění možných metod. V praxi užívané metody většinou kombinují více ze výše zmíněných přístupů. Tím je pak možné zrychlit a zpřesnit detekci.

4.4.1 Detekce na základě barvy

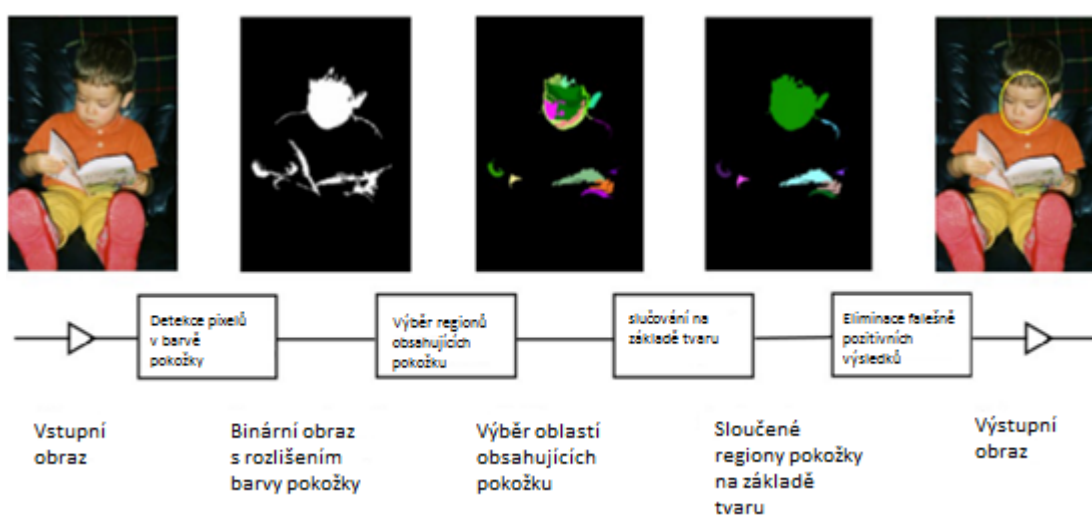
Detekce obličeje na základě jeho barvy je jednou z nejrozšířenějších metod. Tato metoda slouží k detekování oblastí obličeje, nevyužívá se k rozpoznávání obličejů (myšleno detekce konkrétního jedince). Často však bývá mezikrokem.

Základem této metody je předpoklad, že barva obličeje, a lidské kůže obecně, je natolik specifická a invariantní, že je možné ji využít pro detekci v obraze. Nicméně je nutné provádět

tuto detekci na specifických barevných modelech, respektive specificky, pro každý barevný model. Nejčastěji užívané barevné modely pro tuto metodu jsou YCrCb, HSI a RGB.

Každý z těchto modelů má své výhody a nevýhodu a dle toho je třeba k nim přistupovat. Nicméně podle provedených studií je pomocí algoritmů založených na detekci barvy, možno dosáhnout zhruba 95% přesnosti. Příklad algoritmu je znázorněn na obrázku 4.5.

Stejně jako u ostatních typů detekcí i zde se setkáváme s nevýhodami, jež plynou ze změny osvětlení, či při sejmutí obrazu s kamery. Některé nevýhody je možné kompenzovat daným barevným modelem (YCrCb model je do značné míry imunní proti změně jasu). Na možnosti jednotlivých barevných modelů bylo provedeno velké množství prací a experimentů. Obecně se nejčastěji využívá modelu HSI případně YcrCb. [4]



Obrázek 4.5: Schéma detekčního řetězce založeného na detekci barvy[7]

4.4.1.1 Barevné modely

RGB barevný model se skládá ze tří základních barev, jež jsou určeny k aditivnímu smísení, tedy red, green a blue. Tento model je možné si představit jako trojdimenzionální krychlový prostor, kde na jednotlivých osách jsou hodnoty odpovídající červené, modré a zelené, přičemž 0 se udává do počátku a bílá je v opačném rohu krychle. Tento model je hojně využíván v počítačích, nicméně má své nevýhody. Jednotlivé barevné složky jsou značně vzájemně korelované.

YCrCb barevný model je složen rovněž ze 3 složek. Byl navržen především pro snímání a práci s videem. Spolu s dalšími modely jako jsou YUV či YIQ patří do rodiny televizních vysílacích barevných modelů. Nicméně YCrCb je model určený pro digitální data. Základem tohoto modelu je to, že z prostoru RGB je oddělena informace o luminanci a chrominanci. Základní představa tohoto modelu nepracuje s daty (u 8 bitových systému) jež se pohybují v rozmezí 0 – 255. Luminanční složka Y má rozpětí 16 – 235 a chromatické složky Cr a Cb

v rozmezí 16 – 240. Nicméně se mnohdy používá ekvivalentu, kdy jsou hodnoty rozprostřeny po celém intervalu 0 – 255.

HSI barevný model se rovněž skládá ze tří složek. H (hue) barvy, S (saturation) saturace a I(intensity) intenzita. Tento model bývá reprezentován cylindrickými koordináty. Kde H je reprezentováno úhlem 0 – 360°, saturace S odpovídá radiu v rozmezí 0 – 1 a intenzita se pohybuje po ose z kde 0 je černá a 1 je bílá. [1] [11]

4.4.1.2 Detekce v oblasti RGB

Tato metoda předpokládá, že barva lidského těla se mění jako funkce relativního směru k osvětlení. Body odpovídající barvě těla mohou být detekovány za pomoci normalizovaného barevného histogramu a je možné je také normalizovat pro změnu intenzity. Tím je vektor [R, G, B] převeden na [r, g] – vektor normalizovaný barevný vektor. Tak získáme region barvy odpovídající obličejí, jež je užít k detekci. [1]

4.4.1.3 Detekce v oblasti YCrCb

Tento model je méně citlivý na změnu jasové složky, protože barevnost je zde uložena v Cb (modrá) a Cr (červená) a jas Y je uložen zvlášť. Detekce barvy je pak určena na základě správně zvolené prahovací hodnoty pro Cb a Cr. Hodnota Cb a Cr potřebná pro prahování je většinou volena dopředu. Je možné nejprve získat hodnoty s velkého počtu obrazů a tyto hodnoty následně použít, případně použít hodnoty z jiných prací. Výhodou této metody je rovněž to, že barva obličeje zůstává víceméně konstantní, ale mění se jasová složka. Hodnoty barev se rovněž příliš neliší u jednotlivých ras. [1]

4.4.1.4 Detekce v oblasti HSI

V tomto barevném modelu je postup obdobný jako u modelu YCrCb. Rozdíl je v tom, že se nastavují hodnoty H – barva a S saturace. Hodnoty, se stejně jako u předchozího modelu, získají dopředu. Na základě předem získaných hodnot je pak nastavena hranice S a H, a pokud bod spadá do daného intervalu, je vyhodnocen jako odpovídající kůži. [1]

4.4.1.5 Detekční algoritmy

Při praktické implementaci je možné využít jednoho barevného modelu, nebo více barevných modelů najednou. Obecně je lepší využít více modelů, neboť je pak možno zpřesnit detekci. Je však třeba mít při kombinaci více modelů na paměti, výhody a nevýhody každého jednoho z nich. Například barevný model RGB bude citlivější na změnu intenzity osvětlení, neboť u něj není oddělena jasová složka od složky barevné, jako tomu je v případě dalších dvou zmíněných modelů. Rovněž je možné využít i další barevné model mimo těch, jež jsou zmíněny zde.[1]

4.4.2 Analýza hlavních komponent

Další metodou, jež je využívána při detekci a rozpoznávání obličeje je tzv. analýza hlavních komponent. PCA je statistická metoda spadající do širokého oboru faktorové analýzy.

Jejím hlavním účelem je redukce rozměrnosti datového prostoru na menší vnitřní rozměrovost hlavního prostoru (feature space), aby bylo možné popsat data co nejekonomičtěji.

Pomocí PCA je možné odstraňovat redundance, predikovat hodnoty, extrahovat znaky, komprimovat data atd. Metoda pracuje hlavně v lineární prostoru, proto je vhodná pro úpravu obrazu či signálu. Sem spadá tedy i rozpoznávání obličejů. Hlavní ideou při použití PCA při rozpoznávání obličejů je vyjádření 1D vektoru (zkonstruovaného z 2D obrazu) v hlavním prostoru (feature space) tvořeném hlavními komponenty. [7]

4.4.2.1 *Matematický popis PCA*

2-D obraz je možné vyjádřit jako 1-D vektor, kde jsou jednotlivé řádky (případně sloupce) naskládány za sebe. Tím získáme jeden dlouhý vektor, který reprezentuje celý obraz. Takový to obraz je možné vycentrovat na střed (mean centered) a to tak, že nejprve vypočteme průměr.

$$m = \frac{1}{M} \sum_{n=1}^M x_i \quad (4.14)$$

A následně vycentrujeme obraz na střed

$$w_i = x_i - m \quad (4.15)$$

Cílem je nalezení souboru e_i , který má nejdelší na každý w_i . Chceme nalézt soubor M ortogonálních vektorů e_i

$$\lambda_i = \frac{1}{M} \sum_{n=1}^M (e_i^T w_n)^2 \quad (4.16)$$

pro něž množství je maximální s ortonormálním omezením

$$e_l^T e_k = \delta_{lk} \quad (4.17)$$

4.4.2.2 *Rozpoznání obličejů pomocí PCA*

Jakmile jsou vypočteny vlastní vektory, je možné přikročit k rozpoznávání obličejů. PCA počítá báze prostoru, které jsou reprezentovány vektory z trénovací skupiny. Tyto báze vektory, vlastní vektory, jsou ve směru největší odchylky vektorů z trénovací skupiny. Nazýváme je eigenfaces. Na každý eigenface se dá pohlížet jako na určitou vlastnost. Když je daná tvář zobrazena do obličejového prostoru (FS – face space), její vektor v obrazu popisuje významnost jednotlivých rysů obličejů. Daná tvář je ve FS vyjádřena pomocí koeficientů vlastních vektorů.

Při trénování je každá tvář z trénovací skupiny převedena do FS a její komponenty jsou uloženy do paměti. FS musí být vyplněn těmito tvářemi. Zkoumaný obraz je předložen systému a promítnut do FS. Systém vypočítá vzdálenost od všech uložených obličejů. Následně můžou v zásadě nastat čtyři možnosti

- Zkoumaná tvář je blízká prostoru FS a rovněž je blízká konkrétnímu obličej – tím rozpoznáme daný konkrétní obličej
- Zkoumaná tvář je blízká prostoru FS, ale již ne konkrétnímu obličej. – identifikace obličejové struktury, ale nikoliv konkrétního obličeje
- Zkoumaný obraz je vzdálen od FS, ale je blízko třídy tváří (face class) – neobličejová struktura
- Zkoumaný obraz je vzdálen od FS i třídy tváří(face class) – neobličejová struktura

Jestliže je daný obličej dobře reprezentován vFS, pak při jeho rekonstrukci vzniká malé množství chyb. Pokud jde však o obraz odlišný při rekonstrukci, vzniká větší množství chyb. Na základě velikosti chyby se určuje, zda je obraz ještě tváří či nikoliv podle přednastaveného prahu. Další práh pak určuje, zda jde o konkrétní obličej porovnáním s databází (jsou-li dostatečně podobné. [3])

4.4.3 Neuronové sítě

K detekci obličeje je možné využít také neuronové sítě. Zde je ale důležité zvolit vhodnou trénovací skupinu, neboť pravidla, podle nichž je pak rozhodováno, zda je o obličej či nikoliv, je třeba síť naučit dopředu. Důležité jsou ovšem nejen příklady trénovací skupiny s obličej, ale také s neobličejovými strukturami. Existuje velké množství přístupů, jak neuronové sítě implementovat, ať už v procesu učení nebo při samotné detekci. Je možné využívat k detekci pouze jedné neuronové sítě, nebo kombinovat výstupy z více sítí, čím by se mělo dosáhnout větší přesnosti. [7]

4.4.3.1 *Příklad využití neuronových sítí při detekci obličeje*

Příkladem užití neuronových sítí je práce, kterou napsali Henry A. Rowley, Shumeet Baluja, a Takeo Kanade [24]. Ti využívají výhody propojení několika sítí k detekci frontálně posazeného obličeje v černobílém obraze. Jejich metoda pracuje dobře v rozdílných světelných podmínkách. V základu jejich systém funguje následovně. Obraz, který je síti předložen, prochází skrze pre-processing, který má za úkol vylepšit možnosti klasifikace. Patří sem například vyrovnaní hodnoty intenzity osvětlení u pixelů, jež jsou ve zkoumaném okně, vyrovnaní histogramu k rozšíření rozpětí intenzit, tak aby bylo možné kompenzovat rozdíly v osvětlení či kamerovém systému. Dále obraz projde přes neuronovou síť. Ta se skládá z několika vrstev skrytých základních jednotek, kde 4 hledají v podobrazu 10x10 pixelů, 16 v podobrazu 5x5 pixelů (detekce očí a nosu) a 6 v podobrazu překrývajících se pruhů ve velikosti 5x20 (pro detekci úst). Výstupem je prostá informace o přítomnosti, či nepřítomnosti obličeje.

Celá síť je trénována jak na datech obsahujících obličeje, tak na obrazech, jež obličeje postrádají. Trénování využívá bootstrap metodu, která zjednodušuje trénování v tom smyslu, že falešné detekce jsou přidány do trénovací skupiny během průběhu. Touto metodou se snižuje objem trénovací skupiny a celý trénovací proces se zrychlí. Aby klasifikátory příliš nezávisely na změnách velikosti, rotaci a posunu, trénovací skupina je rozšířena o rotace (10°), změny

ve velikosti (90 – 110 %) a posuvu (polovina pixelu). Síť je nejprve náhodně nastavena a poté je trénována na výstup 1 pro obličejové struktury a -1 pro struktury neobličejové.

U příkladů s jednou neuronovou sítí bylo dosaženo přesnosti výsledku přes 90%, ale s velkým počtem falešně pozitivních obrazů. Pro vylepšení je možné využít více sítí najednou. Propojení překrývajících se detekcí uchová části, kde byl počet detekcí vysoký a eliminuje ty, kde byl malý. Tento model selhává v malém počtu příkladů, například při v případě, kdy jedna tvář zakrývá druhou.

Při využití více sítí je ale třeba užít zastupování mezi jednotlivými sítěmi. Pak je možné získat výsledek z více sítí najednou. Je-li užita logika AND, sníží se detekce obličejů na 78%, ale výrazně se omezí počet falešných detekcí. Při užití logiky OR je detekce zhruba na 90% ovšem s opětovným zvýšením falešných detekcí. Další komplexní metody propojení sítí pak dosahují asi 84% detekce s přijatelným množstvím falešně pozitivních detekcí.

4.4.4 Detekce obličeje pomocí Haar klasifikátorů

Tato metoda detekuje obličej, případně jednotlivé rysy, jako jsou oči, nos či ústa. Základem je přitom soubor Haarovy příznaky (Haar-likefeatures). Tato metoda nevyužívá hodnot intenzity u jednotlivých pixelů ale lokální změny v kontrastu mezi obdélníkovými skupinami sousedících pixelů. Těmito skupinami pixelů se detekují lokální odchylky ve změně jasu a podle nich se pak určují světlé a tmavé oblasti. Dvě či tři přilehlé oblasti s relativní odchylkou v kontrastu pak tvoří Haarovy příznaky. Ty jsou pak využívány k detekci obrazu. U Haarových příznaků může být snadno měněna velikost a to tak, že se zahrne více či méně pixelů do jednotlivých zkoumaných skupin. Tak je možné detekovat objekty různé velikosti.

Základem je vytvoření tzv. integrálního obrazu. Pracuje se v základu s jednoduchými obdélníkovými obrysy, z nichž se získá integrální obraz. Ten je vypočten podle rovnice

$$AI[x, y] = \sum A(x', y') \quad (4.18)$$

Jde v podstatě o pole obsahující sumu intenzit pixelů umístěných vlevo a nahoře od dané pozice. Je možné podobně vypočítat i integrální obraz otočený o 45°. Je potřeba pouze dvou výpočtů pro obě tato pole. Přičemž nezáleží na tom, jak velkou oblast budeme chtít obsáhnout. Detekce různě velkých oblastí je v zásadě stejně početně náročná. Nicméně je třeba zmínit, že pro aplikaci tohoto algoritmu je také potřeba získat menší oblast, která bude s největší pravděpodobností obsahovat požadované rysy. Rovněž, pokud budeme pracovat například s podobrazem velikosti 24x24 pixelů výpočet všech možností mezi jednotlivými pixely, by bylo nepraktické. Proto se postupuje kaskádně. Využívá se co nejmenšího nutného počtu rysů k definování obrazu. Pokud jsou klasifikátory správně poskládány do kaskády, je možné zkoumat pouze tu část obrazu, která s největší pravděpodobností obsahuje hledaný objekt a v jednotlivých úrovních detekci upřesňovat. Podle počtu úrovní, na kterých jsou prováděny výpočty, také můžeme zvýšit či snížit přesnost.

Tato metoda však, podobně jako jiné metody, vyžaduje, aby byly klasifikátory nejprve získány a to pomocí trénování. K trénování se využívá AdaBoost algoritmus a algoritmus pro tvorbu Haarových klasifikátorů. Obě tyto metody se dají implementovat samostatně, nebo je možné využít jich již z open source knihovny opneCV. Tato knihovně rovněž obsahuje i již vytvořené a předtrénované kaskády pro detekce nejběžnějších rysů. Pokud chceme provést vlastní trénování, je třeba si vytvořit dvě série obrazů. Jednu kde není přítomen daný rys (obecně to můžou být jakékoliv fotografie přírody, jednoduchých obytných prostor, měst či další. Tyto obrazy pak musí být nastaveny jako negativní příklady. Pozitivní skupina pak musí obsahovat hledaný rys a dopředu se určuje umístění a velikost hledaného rysu. Následně je pomocí zmíněných algoritmů vytvořena kaskáda pro detekci.

Podle provedených studií je možné touto metodou dosáhnout poměrně přesných výsledků i při přijatelné výpočetní rychlosti. Rovněž je možné detekovat i předměty pod různými úhly. Tato metoda je méně citlivá na pozici než jiné využívané metody, které často pracují pouze s frontálním pohledem. Nicméně má v tomto také svá omezení a hledaný rys musí být v obraze vidět.

5 Návrh řešení

Pro správné řešení je nejprve potřeba vyhodnotit problémy, které bude třeba překonat, aby byla detekce osob co nejpřesnější. Nejprve bude dobré problém zjednodušit na detekci obličeje tedy, kolik bude v obraze obličejů, tolik bude v automobilu osob.

Zde se ovšem objevují stejné problémy, jako při jiných detekcích obličeje. Sem spadají detekce obličejů v zákrytu (šátek, brýle), detekce obličeje v jiné než frontální pozici, detekce obličeje v případě grimasy. Existují řešení těchto problémů, ale s nimi zase narůstá složitost výpočtu. Je tedy třeba zvolit metodu, která bude podávat nejlepší výsledku v poměru rychlost/výkon.

Některé metody detekce objektů v obraze (například Houghova transformace) jsou výpočetně velice náročné a proto by bylo dobré co nejvíce snížit objem dat, které takto bude třeba zkoumat. Tedy nalézt nejprve objekty, které jsou podezřelé z toho, že se jedná o obličej.

Pro tento postup je nejlepší využít specifity v barvě obličeje, která je poměrně nestandardní. Pro další průzkum nám pak mohou zůstat pouze části obrazu, kde se vyskytuje požadovaná barva. U této metody nemusíme řešit problém se slunečními brýlemi (ty nebudou zakrývat celý obličej), ale v případě zakrytí obličeje šátkem nebude tato metoda podávat nejlepší výsledky. Rovněž je třeba počítat s tím, že budeme detekovat všechny oblasti, kde se bude vyskytovat lidská kůže, tedy i ruce a nohy.

Dalším detekčním mechanismem by měla být detekce pohybu, neboť se dá předpokládat, že se obličej na videu budou vždy mírně pohybovat. Tento postup by měl zahrnout i detekci obličeje, zakrytého šátkem, případně jinak. Nicméně však tato metoda není příliš specifická, pohybujících se objektů bude v obraze pravděpodobně více.

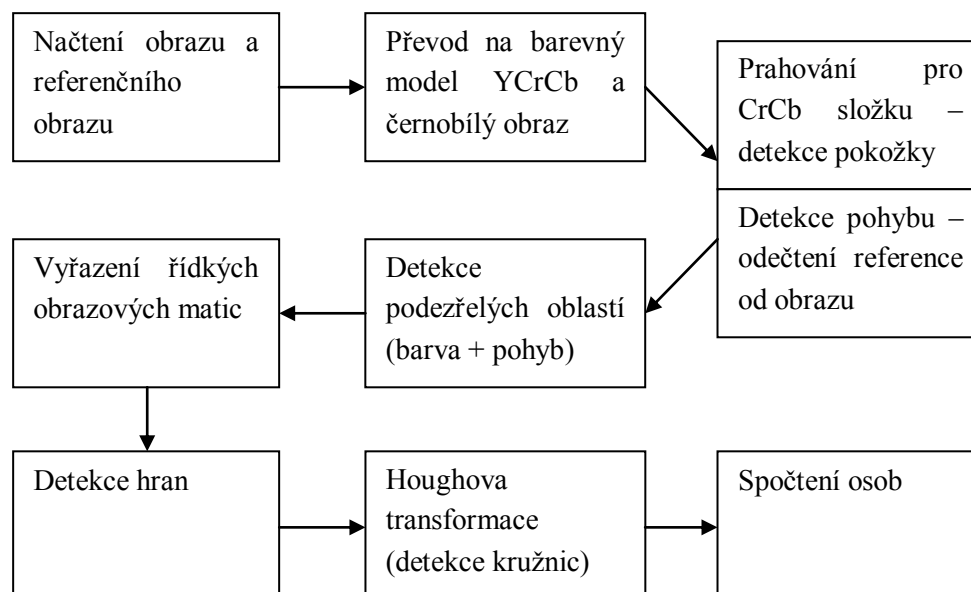
Je však možné předchozí dvě metody zkombinovat. Praktický postup bude podobný jako je například u digitální substrakční angiografie, tedy jeden obraz odečteme od druhého, abychom získali požadovaný rozdíl. Pohybující se části budou na jiných místech. Použitím kombinace metody detekce obrazu a barvy obličeje by mělo být možné redukovat objem dat k dalšímu prozkoumání na nutné minimum a tím snížit délku výpočtu.

Pro potvrzení, že se jedná o obličej, by pak měla být použita vyšší metoda například rozhodovací kaskáda, případně detekce primitiv typických pro obličej (Houghova transformace pro detekci kružnic přítomných v oku). Tyto metody by však již měly být použity na výřezy z původního obrazu, o nichž se domníváme, že obsahují obličej.

Pro získání obrazových matic, v nichž předpokládáme obličej, bude potřeba zjistit, zda se o obličej skutečně jedná. Pokud budeme postupovat na základě barvy, je možné vyloučit části, kde je tato barva přítomná, ale na obličej je jí málo. Náhodné malé části s podobnou barvou, okrajové části obrazu atd.

Pro té bude potřeba odlišit části, jako jsou ruce, nohy, hrudník a další, které mohou být barvou pokládány za obličej. To by mělo být provedeno pomocí detekce očí Houghovou

transformací. Houghova transformace vyžaduje také určité předzpracování obrazu a to především převod na černobílý obraz a následně detekci hran. Pak bude možné použít Houghovu transformaci pro detekci kružnice. Nicméně je možné, že tato metoda nebude vhodná kvůli rychlosti zpracování.



Obrázek 5.1: Blokové schéma navrženého detektoru

5.1 Návrh jednotlivých částí detekčního řetězce

Načtení obrazu a načtení referenčního obrazu by mělo být provedeno pomocí možností knihovny OpenCV, která je pro tyto operace navržena. Obraz by měl být načítán z web kamery s rozlišením 640x480. Prvotní referenční obraz by měl být uložen v paměti detekčního zařízení. Dále by mělo být jako reference využito předchozího snímku. Tedy snímek, který byl využit, jako zkoumaný obraz je na konci cyklu nastaven jako reference a z kamery je načten nový zkoumaný snímek.

Dalším krokem je převedení do barevného modelu YCrCb a rovněž převod na černobílý obraz. Původní snímky jsou uloženy v barevném modelu RGB. Převod do YCrCb bude realizován pouze u zkoumaného obrazu, převod na černobílý obraz pak bude proveden u zkoumaného obrazu i reference. Převod bude realizovaný na základě převodních rovnic získaných z odborné literatury.

Následně bude provedena detekce pohybu a detekce barvy obličeje. Detekce pohybu bude realizována na základě odečtení referenčního snímku od snímku zkoumaného. Dále pak bude pracováno s místy, kde bude detekována změna, ostatní části obrazu budou nastaveny na hodnotu jasu 0 případně 255. Detekce barvy obličeje bude realizována na základě prahování. Práh bude stanoven pro červenou (Cr) a modrou (Cb) složku na základě odborných prací a testování.

Detekce podezřelých oblastí - v tomto kroku půjde o kombinaci obou principů a to jak detekce barvy obličeje, tak detekce pohybu. Místa, která budou odpovídat oběma těmto kritériím, budou vyhodnocena jako místa s největší pravděpodobností výskytu obličeje. Tento model předpokládá, že člověk není statický a že se stále mírně pohybuje a zároveň s tím, že barva kůže je dostatečně specifická, aby bylo možné ji oddělit od okolí. Po detekci těchto míst budou z dané oblasti pořízeny menší obrazové matice, které by mělo v ideálním případě obsahovat lidský obličej.

Dá se předpokládat, že na některých místech bude detekován jen malý počet bodů, jež budou odpovídat oběma kritériím (detekci pohybu i barvy obličeje). Tyto oblasti a výřezové matice by neměly být v souboru ponechány. Proto budou vyřazeny na základě prahování. V dané výřezové obrazové matici budou spočteny všechny body, které odpovídají barvě obličeje. Podle jejich počtu pak bude rozhodnuto, zda je daná oblast dostatečně velká na to, aby obsahovala celý obličej. Práh bude nastaven na základě testování.

Dalším krokem by měla být detekce hran a následná Houghova transformace. Tou by měly být detekovány kružnice odpovídající očím. Na základě jejich detekování by měla být daná výřezová matice vyhodnocena jako obličej, případně vyřazena. Posledním krokem je spočtení zbylých výřezových matic. Ty by již měly obsahovat všechny obličeje a měly by tak odpovídat i počtu osob v automobilu.

5.2 Požadavky na zatížení

Jedním z hlavních omezujících faktorů při výběru vhodné metody detekce a její implementaci je také časové zatížení programu. Tedy jak dlouho bude detekce z jednoho pořízeného snímku trvat. Můžeme počítat s tím, že informace o počtu osob v automobilu může být po nějakou dobu uchovávána v paměti. Nicméně je třeba si uvědomit, že během jízdy se počet osob může změnit (některé osoby vystoupí, případně nastoupí). Proto by detekce měla být prováděna opakovaně. Opakovaná detekce osob také může zamezit chybám v měření. Tedy je možné vypočítat průměr z více měření. Tím by měla klesnout chybovost.

Bylo by dobré, aby program nepracoval s jedním obrazem více než několik sekund. Rovněž by se měl brát ohled na to, jaký prostor bude program sám o sobě zabírat v paměti detekčního zařízení.

6 Řešení

Celý původní program pro detekci osob v automobilu byl napsán v programovacím jazyku C, aby bylo možné jej následně nahrát na vývojovou desku Quickstarboard i.MX53 a tam poté spustit. Program byl napsán a odladěn pomocí vývojového prostředí NetBeans IDE 7.4. Využita byla open source knihovna pro zpracování obrazu OpenCV.

6.1 Knihovna OpenCV

Prvním krokem řešení bylo stáhnutí a využití knihovny OpenCV.

OpenCV (Open Source Computer Vision Library) je open source softwarová knihovna určená pro zpracování obrazu převážně pro počítačové vidění a realtime zpracování. OpenCV bylo navrženo pro vybudování obecných základních systémů pro aplikace pracující s počítačovým viděním, pro jejich zrychlení a zjednodušení jak v soukromé tak komerční sféře.

Knihovna obsahuje více než 2500 optimalizovaných algoritmů, které zahrnují dostatečné množství algoritmů pro klasickou práci i pro nejpokročilejší aplikace. Algoritmy mohou být využity pro detekci obličejů, identifikaci objektů, detekci pohybů, sledování pohybujících se objektů, extrahování 3D modelů objektů, spojování videa pro 3D ze stereo kamer, odstraňování červených očí a mnoho dalších. [13]

Knihovnu používají jak výzkumníci, firmy, tak vládní kruhy. Mezi známé firmy patří například Google, Yahoo, Toyota, Microsoft, Intel atd.

Tato knihovna má algoritmy pro rozhraní s C++, C, Python, Java a Matlab. Přičemž podporuje práci na systémech Windows, Linux, Android a Mac OS. OpenCV je přednostně určeno pro realtime zpracování a počítačové vidění.

V práci byly využity především tyto metody a struktury.

Pro ukládání a zpracovávání obrazu existují dvě základní struktury. IplImage a CvMat, je možné obraz převádět z jedné struktury do druhé a to různými způsoby. V práci je využívána především struktura CvMat, ve které jsou uložena všechna zpracovávaná data. Struktura IplImage je využita pouze při načítání videa a následně je převedena na CvMat.[13]

CvMat je multikanálová hustá matice v níž je obraz uložen v jednorozměrném vektoru (jednotlivé řádky jsou uloženy za sebou). Základní prvky této struktury jsou

- int type
- int step – celková délka řádku v bytech
- int* refcount
- union data – ukazatel na konkrétní databáze
- ptr – ukazatel na 8-bit unsigned elementy
- s – ukazatel na 16-bit signed elementy
- i – ukazatel na 32-bit signed elementy

- fl – ukazatel na 32-bitfloating point elementy
- db – ukazatel na 64-bitfloating point elementy
- introws – počet řádků
- intcols – počet sloupců

Obraz je pomocí OpenCV načten do jednorozměrného pole. Jednotlivé řádky jsou tedy uloženy v řadě za sebou. Tato metoda ukládání dat je výhodná především kvůli rychlosti zpracování, neboť k prvkům jednorozměrného pole je možné přistoupit rychleji, než k prvkům v poli dvourozměrném.

Struktura CvMat, do níž jsou ukládána data, rovněž obsahuje některé další údaje, mimo samotných obrazových dat. Je to především, výška a šířka obrazu, barevná hloubka a další. Hlavní význam má především výška a šířka obrazu, protože na základě těchto údajů pak přistupujeme k jednotlivým prvkům pole, jako by to byla dvojrozměrná matice.

Ve struktuře IplImage je uloženo větší množství informací o datech. Tato struktura je převzata od Intelu z Intel Image processinglibrary. Knihovna OpenCV podporuje jen některé podmnožiny struktury IplImage. [13]

Dále knihovna nabízí i možnosti převodu mezi jednotlivými barevnými modely a převod na černobílý obraz. Je možné využívat různé druhy detektorů hran či obrazových primitiv. Rovněž jsou zde možnosti využití již předpřipravených detektorů obličeje. Nicméně mnohé z těchto funkcí nejsou určeny pro všechny programovací jazyky. Některé funkce mohou být navrženy pouze pro C++ nebo python. Rovněž mohou mít některé nedostatky, případně být chybové. Z knihovny nebyly tyto části v práci využity.

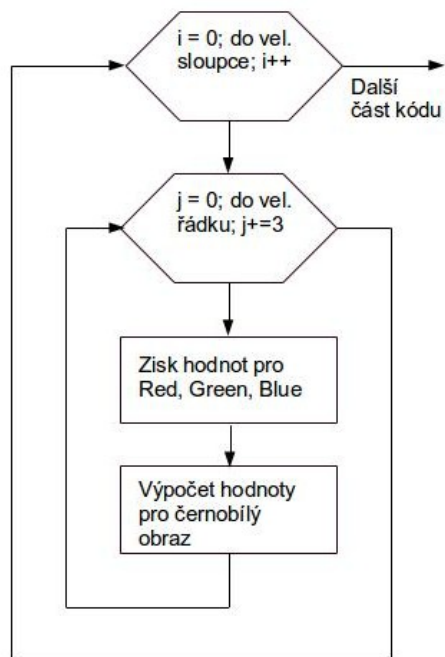
6.2 Implementace navrženého algoritmu

Prvním krokem bylo načtení obrazu a videa. Pro práci se soubory a pro další úpravy byla využita knihovna OpenCV. Video je načítáno z web kamery (velikost obrazu 640x480), přičemž je doba mezi jednotlivými načtenými obrazy zhruba 1s. Tato prodleva je zde jednak proto, že na desce je pomalejší výpočetní rychlost a jednak proto, aby byl mezi jednotlivými snímky určitý časový rozdíl, aby bylo možné bezpečně zaznamenávat pohyb.

6.2.1 Převod na černobílý obraz a převod na YCrCb barevný model.

Před detekcí obličeje je třeba zkoumaný i referenční obraz převést na černobílý, pomocí něhož je pak detekován pohyb. A také do barevného prostoru YCrCb, kde je detekována barva obličeje.

Pro převod na černobílý obraz je využita následující rovnice 6.1. Je dobré zmínit, že v poli jsou hodnoty barvy uloženy obráceně, tedy ne v pořadí Red Green Blue, ale Blue Green Red. Což je také možné pozorovat v kódu níže. Černobílý obraz je vypočten jak pro referenční obraz, tak pro obraz zkoumaný. Oba jsou uloženy ve struktuře CvMat a dále zpracovány při detekci pohybu. Na obrázku 6.1. je uveden vývojový diagram odpovídající části algoritmu, dále pak příklad implementovaného kódu.



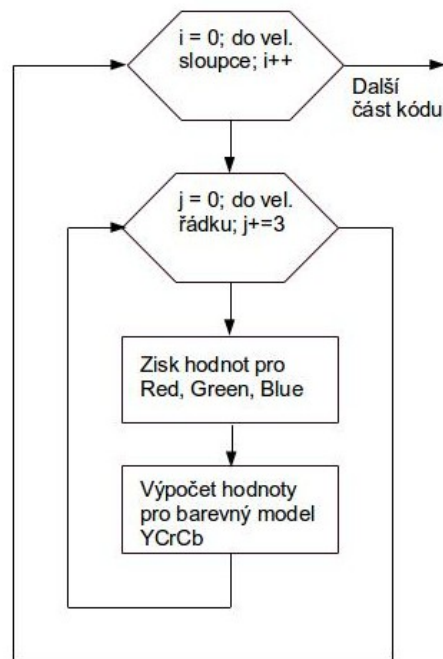
Obrázek 6.1: vývojový diagram převodu z RGB na černobílý obraz

$$Y = (0,299 \times R) + (0,587 \times G) + (0,114 \times B) \quad (6.1)$$

```

for (i = 0; i < radek; i++) {
for (j = 0; j < sloupec * 3; j += 3) {
bl = obraz->data.ptr[i * sloupec * 3 + j];
    g = obraz->data.ptr[i * sloupec * 3 + j + 1];
    r = obraz->data.ptr[i * sloupec * 3 + j + 2];
    cernobila->data.ptr[i * sloupec + j / 3] = (0.299 * r) + (0.587 *
g) + (0.114 * bl);
    }
}
  
```

Převod do barevného prostoru YCrCb je obdobný. Zde je ale potřeba využít tří rovnic (6.2); (6.3); (6.4), abychom mohli vypočítat všechny tři složky. Zde jsou pro nás nejdůležitější složky Cr a Cb, které jsou následně využity pro detekci obličeje a to pomocí správně zvoleného prahu. Přepočet je programem proveden pomocí těchto tří rovnic (6.2),(6.3),(6.4). Na obrázku 6.2 je uveden vývojový diagram příslušné části kódu.



Obrázek 6.2: Vývojový diagram převodu z RGB do YCrCb

$$Y = (0,299 \times R) + (0,587 \times G) + (0,114 \times B) \quad (6.2)$$

$$Cb = (-0,168 \times R) - (0,331 \times G) + (0,499 \times B) + 128 \quad (6.3)$$

$$Cr = (0,499 \times R) - (0,418 \times G) - (0,0813 \times B) + 128 \quad (6.4)$$

```

for (i = 0; i < radek; i++) {
    for (j = 0; j < sloupec * 3; j += 3) {
        b1 = obraz->data.ptr[i * sloupec * 3 + j];
        g = obraz->data.ptr[i * sloupec * 3 + j + 1];
        r = obraz->data.ptr[i * sloupec * 3 + j + 2];

        Y = (((0.299 * r) + (0.587 * g) + (0.114 * b1) + 0));
        YCrCb->data.ptr[i * sloupec * 3 + j] = Y;

        Cr = (((0.499 * r) - (0.418 * g) - (0.0813 * b1) + 128));
        YCrCb->data.ptr[i * sloupec * 3 + j + 1] = Cr;

        Cb = (((-0.168 * r) - (0.331 * g) + (0.499 * b1) + 128));
    }
}

```

```
YCrCb->data.ptr[i * sloupec * 3 + j + 2] = Cb;
```

Obraz je uložen do struktury CvMat a dále zpracován. Pro převod existuje velké množství rovnic, většinou však pomocí nich nedosáhneme hodnot v rozmezí 0 – 255. Správně by tomu tak ani nemělo být. Tento model však zajišťuje, že vypočtené hodnoty budou v tomto rozsahu. Hodnoty pro YUV by však měly mít správně o něco nižší rozsah. Nicméně přesnost, kterou dosáhneme těmito rovnicemi je dostačující, a práce s tímto barevným modelem se podstatně zjednoduší. Experimentálně byla funkčnost těchto rovnic ověřena.

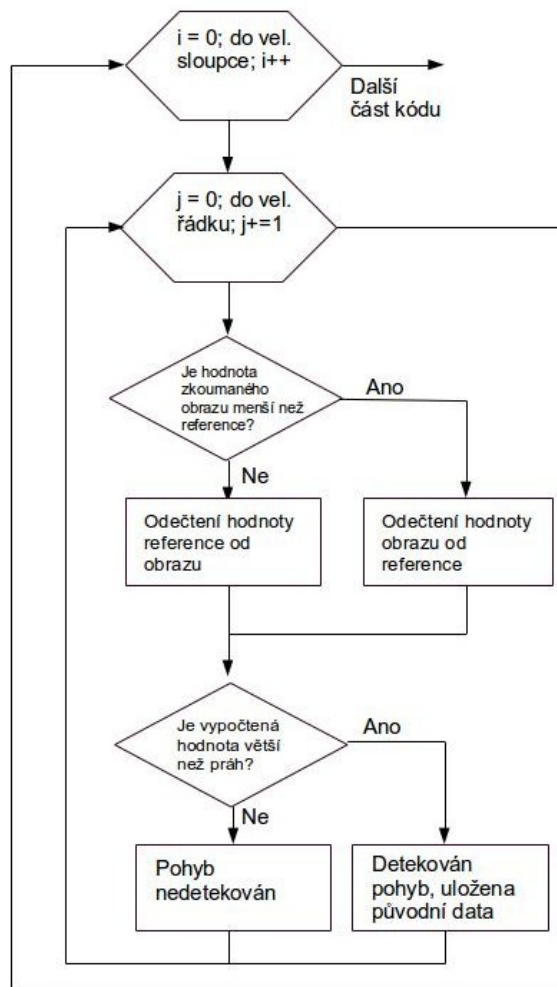
6.3 Detekce pohybu

Pohyb je detekován na základě porovnání současného snímku se snímkem předchozím. Všechny obrazy jsou uloženy do struktury CvMat, jež je využívána knihovnou OpenCV. Samotná struktura je popsána výše. První získaný snímek je porovnán se snímkem referenčním, který je uložen v paměti desky. Po výpočtu je vždy současný snímek nastaven jako referenční a je s ním porovnán snímek aktuální.

Před detekováním pohybu jsou využity černobílé snímky. Černobílé snímky jsou využívány v programu i jinde a práce s nimi je rychlejší a jednodušší než s barevnými modely. Pro detekci pohybu se jeví jako nejlepší.

Prakticky je pohyb detekován tak, že jsou porovnány odpovídající body matice a následně jsou od sebe odečteny (vždy je nižší hodnota odečtena od vyšší, aby byla získána absolutní hodnota odchylky). Následně je hodnota prahována, a pokud odchylka nižší než hodnota 10 (hodnoty se pohybují v klasickém rozsahu 0 – 255), je daný bod nastaven na bílou (255). Ve snímku jsou pak ponechány pouze hodnoty, u kterých je změna větší. Dá se předpokládat, že v obraze zůstanou pouze ty předměty, které se pohybují a tak bude rozdíl dostatečně velký. Jinak by se měly na sebe zobrazovat body s přibližně stejnými hodnotami. Na obrázku 6.3 je uveden vývojový diagram použitého algoritmu a dále výňatek z implementovaného kódu.

Během testování se jako nedostatek prokázalo to, že se může stát, že za obličejem se může nacházet plocha se stejnou barvou, pak není daná plocha detekována. Nicméně jsou většinou detekovány alespoň hrany obličeje, což je v tomto případě dostačující.



Obrázek 6.3: Vývojový diagram detekce pohybu

```

for (i = 0; i < radek; i++) {
    for (j = 0; j < sloupec; j++) {
        if (cernobila->data.ptr[i * sloupec + j]
            > cernobilaref->data.ptr[i * sloupec + j]) {
            result = cernobila->data.ptr[i * sloupec +
                j] - cernobilaref->data.ptr[i * sloupec +
                j];
        }
        else result = cernobilaref->data.ptr[i * sloupec
            + j] - cernobila->data.ptr[i * sloupec + j];
        if (result < 10) {
            motion->data.ptr[i * sloupec + j] = 255;
        }
    }
}

```

```

    }
    elsemotion->data.ptr[i * sloupec + j] = result;

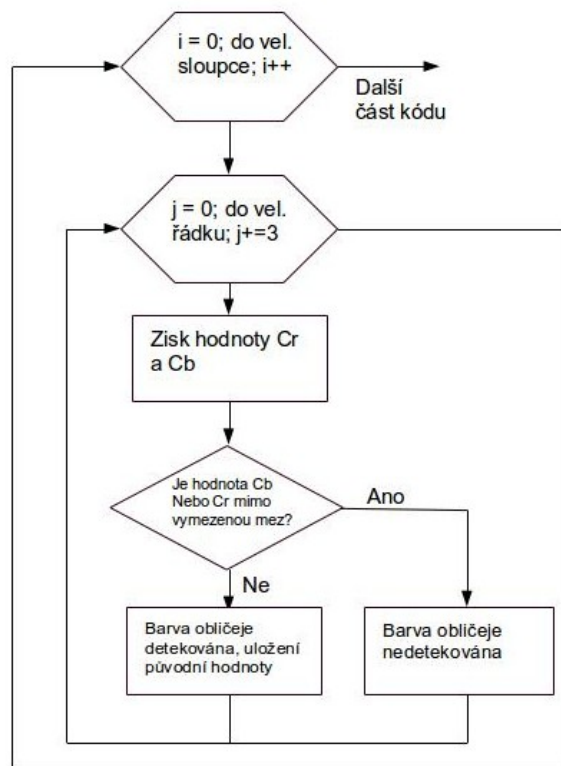
```

6.3.1 Detekce barvy obličeje

V této práci využito detekce obličeje na základě barvy a to v barevném prostoru YCrCb. Tento model byl zvolen, protože dle výzkumů[1] je pro detekci barvy vhodnější, neboť je zde oddělena jasová složka Y od složek barevných Cb (modrá složka) a Cr (červená složka). Tímto by mělo být dosaženo menší závislosti na změně osvětlení.

Pomocí experimentů a měření na větším počtu obrazů byl několika výzkumníky odhadnut práh pro Cr a Cb složky, pomocí níž se zachytí barva, odpovídající lidské kůži. Obecně se barva kůže nemění, mění se pouze jasová složka. Ta zde ale nehraje velkou roli a proto je možné takto obličej detekovat i při významných změnách jasu.

Hodnoty prahu vycházely z práce [1] a testování a jsou uvedeny v (6.5) a (6.6). Nicméně hodnoty se liší od hodnot uvedených v odborném zdroji [1], během testování byly poupraveny, aby co nejvíce odpovídaly daným podmínkám. Změněny byly hodnoty hlavně modré složky Cb. [11]



Obrázek 6.4: Vývojový diagram detekce barvy obličeje

$$100 \leq Cb \leq 125 \quad (6.5)$$

$$135 \leq Cr \leq 155$$

(6.6)

```
for (i = 0; i < radek; i++) {
    for (j = 0; j < sloupec * 3; j += 3) {
        Cr = YCrCb->data.ptr[i * sloupec * 3 + j + 1];
        Cb = YCrCb->data.ptr[i * sloupec * 3 + j + 2];
        if (Cb < 100 || Cb > 125) {
            prahovaniCrCb->data.ptr[i * sloupec * 3 + j] = 0;
            prahovaniCrCb->data.ptr[i * sloupec * 3 + j + 1] = 0;
            prahovaniCrCb->data.ptr[i * sloupec * 3 + j + 2] = 0;
        }
        if (Cr < 135 || Cr > 155) {
            prahovaniCrCb->data.ptr[i * sloupec * 3 + j] = 0;
            prahovaniCrCb->data.ptr[i * sloupec * 3 + j + 1] = 0;
            prahovaniCrCb->data.ptr[i * sloupec * 3 + j + 2] = 0;
        }
    }
}
```

prahovaniCrCb – jde o matici, v níž je uložen klasický barevný model RGB, ale jsou v něm ponechány pouze hodnoty, které leží v místě, kde jsou odpovídající hodnoty modré Cb a červené Cr složky. Ostatní hodnoty jsou nastaveny na 0 (černá). Příklad je možno vidět na obrázku níže.

6.4 Detekce oblastí podezřelých z přítomnosti obličeje

Získaná data z detekce pohybu a z detekce barvy obličeje jsou prokombinována tak, že jsou dále brány v úvahu pouze body, pro které platí obě detekce. Tedy body, jež se pohybují a zároveň mají také barvu odpovídající barvě obličeje. Na základě těchto bodů jsou provedeny výřezy z původní obrazové matice, které mají velikost v poměru 160x240 bodů. Tím jsou podstatně menší, než je celá původní matice.

Nedá se předpokládat, že v obrazu budou detekovány pouze body odpovídající obličeji. Ve scéně se mohou nacházet i další, avšak podstatně menší objekty, které budou odpovídat jak barvou, tak drobným pohybem. Pokud však v daném výřezu spočteme počet hodnot, jež odpovídá dané barvě, jsme schopni tyto drobné objekty, či náhodné body eliminovat. Proto program spočítá, kolik bodů, jež odpovídají svou barvou, se nachází ve výřezu. Pokud tato hodnota přesáhne nastavený práh (8000 bodů) je obraz vyhodnocen, jako oblast obsahující obličej. Vzhledem k tomu, že se může stát, že obličej nebude ve výřezu celý (první detekovaný bod nemusí být přímo bodem obličeje, ale může ležet v jeho blízkosti). Je následně v okolí této matice vymazán větší objem dat. Tedy detektor pohybu v oblasti okolo daného výřezu je

nastaven na 255. Při hledání další podezřelé oblasti je pak vymazaná oblast vynechána. Tím se sníží pravděpodobnost, že obličej bude detekován dvakrát.



Obrázek 6.5: Vývojový diagram určení zda je podezřelá oblast obličejem

```

if (l > 8000) {
    oblicej++;
    for (l = i - (80 * kon); l < i + (260 * kon); l++) {
        for (k = j - (60 * kon); k < j + (160 * kon); k++) {
            if((l*sloupec + k) < (radek*sloupec)&&(l*sloupec
            + k) > 0)motion->data.ptr[l * sloupec + k] =
            255;
        }
    }
}

```

Pokud je hodnota prahu překročena, je rovněž zvýšen počet obličejů a tedy i počet osob detekovaných v automobilu. Pro ilustraci jsou níže uvedeny výřezy z matic, které byly pořízeny při testování.

6.5 Houghova transformace

Jako metoda pro další potvrzení, že nalezená oblast je obličejem, měla být využita metoda Houghovy transformace. Proto byl na vyřezané matice použit Cannyho detektor, neboť Houghova transformace pracuje s binárním vstupním obrazem. Cannyho detektor byl využit z funkcí knihovny OpenCV. Byl otestován i klasický Sobelův detektor aplikovaný všesměrově, ale nepodával zdaleka tak kvalitní výsledky jako Cannyho detektor.

Následně byl rovněž proveden, pokus o použití detekce kruhů pomocí Houghovy transformace, jež nabízela knihovna OpenCV, nicméně tato metoda nefungovala správně. Proto byla HT následně naprogramována samostatně. S využitím parametrických rovnic uvedených výše. Nicméně při testování bylo zjištěno, že metoda je velice pomalá (řádově desítky sekund až minuty). Byly provedeny pokusy o optimalizaci a to výrazným snížením poloměru detekovaných kruhů, neboť poloměr kruhu byl zhruba znám na základě velikosti očí. Nicméně tato hodnota není známa přesně, proto stále musel být použit určitý interval. Tímto se výpočetní

náročnost značně snížila a HT proběhla řádově v sekundách. Nicméně byla tato doba výpočtu pořád příliš velká. Pokud by bylo k prozkoumání ponecháno 5 výřezových obrazových matic délka výpočtu by značně vzrostla a to pro daný program není přípustné. Proto bylo rozhodnuto tuto metodu potvrzení vynechat.

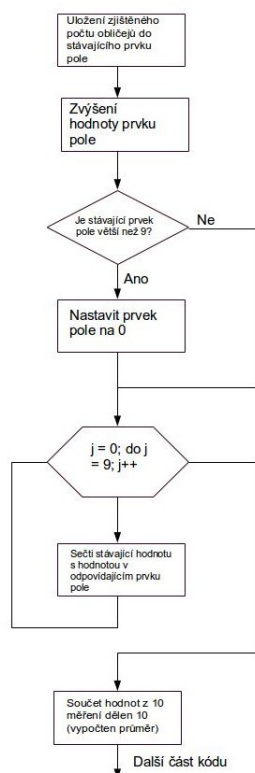
Jako další možnost se pak jeví implementace Haarovy kaskády, kterou rovněž nabízí knihovna OpenCV, a kterou je možno využít pro detekci očí, úst a dalších částí obličeje a také pro celý obličej, nicméně tuto metodu se nepodařilo prozatím zprovoznit. Rovněž je možné, že i tato metoda by nebyla dostatečně rychlá. Další metodou by mohla být detekce na základě textury obličeje. Tuto metodu by bylo dobré vyzkoušet.

Další metodou by mohl být také genetický algoritmus pro detekci kruhů, či jiné metody pro detekci obličeje, které však vyžadují využití databáze s příklady obličejů, případně i neobličejových částí.

6.6 Upřesnění výsledku

Protože nebylo možné využít Houghovu transformace pro upřesnění výsledku, bylo zpřesnění provedeno alespoň následovně. Celý detekční cyklus byl prodloužen na dobu 10s a výsledná hodnota počtu osob v automobilu je získána jako průměr z 10 posledních hodnot. Výsledek je tedy možné získat po 10s od zpuštění programu. Tímto je možné eliminovat náhodné nedostatky v měření a zpřesnit tak detekci. Pokud je méně než 5 měření z 10 vyhodnoceno špatně (s rozsahem +/- 1 obličej). Mělo by být pořád možné hodnotu detekovat přesně. Na obrázku 6.6 je uveden vývojový diagram.

```
pocetoblliceju[oblliceje] = obllicej;
oblliceje++;
if (oblliceje> 9) oblliceje = 0;
    k = 0;
for (j = 0; j < 9; j++) {
    k = k + pocetoblliceju [j];
}
obllicej = k / 10;
```



Obrázek 6.6: Vývojový diagram pro upřesnění výsledku

6.7 Časová informace

Ke zprávě o počtu osob by rovněž měla být přidána informace o čase, kdy došlo k nehodě. Informace je ve formátu „den v týdnu; měsíc; den; aktuální čas hh:mm:ss; rok“. Tato informace se na obrazovku vypíše při každém vypsání počtu detekovaných osob.

6.8 Ukončení programu

Celý program běží ve smyčce, tak dlouho, dokud nejsou požadovány hodnoty k odeslání. Tedy počet osob a v tomto případě i čas a datum. Pro vypnutí je třeba ukončit program pomocí ctrl+C a to jak na vývojové desce, tak u programu v počítači. Po ukončení programuse na obrazovku vypíše, kolik osob je v automobilu detekováno. Rovněž je k počtu osob i přiřazena informace o čase, kdy byl program ukončen (teoreticky, kdy došlo k nehodě). Tyto informace pak mohou být dále využity při odesílání sms na dispečink záchranné služby. Ve vytvořeném programu se po ukončení také uloží některé ilustrační obrázky, černobílé obrazy, detekce pohybu, detekce barvy kůže, jednotlivé vyřezané obrazové matice a další. Toto ukládání se provede pouze po ukončení hlavní smyčky a nezatěžuje tak program během běhu. V programu na desce k ukládání těchto dat nedochází. Pro detekci nejsou potřeba a jde pouze o ilustrační materiál. Na vývojové desce část s ukládáním dat chybí.

6.9 Quickstarboard i.MX53

Deska, která je využita pro detekci počtu osob z automobilu je od firmy freescale a je navržena i pro zpracovávání videa.

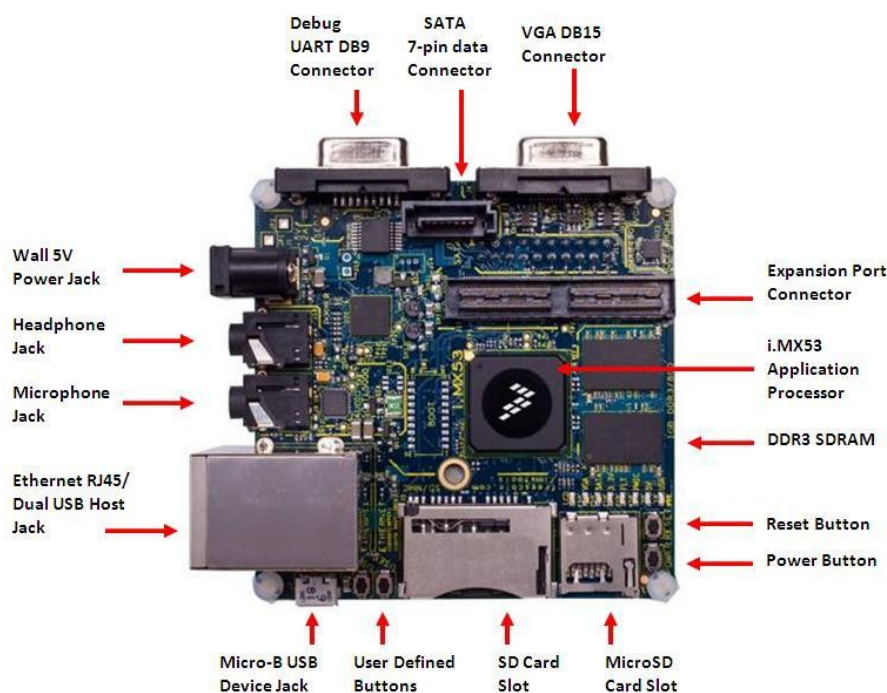
Deska i.MX53 Quick Start Board je určena pro multimediální aplikace na procesorech ARM Cortex™-A8. Platforma obsahuje schéma hardwarového zapojení a podporuje package (BSP) obsahující ovladače určené pro muldimediální operace.

Deska obsahuje

- i.MX53 applicationprocessor - ARM® Cortex™-A8 s pracovní rychlostí až do 1,2GHz.
- Freescale MC34708 PMIC
- 1 GB DDR3 SDRAM memory
- Dual USB host konektor - je možné k desce připojit klávesnici a myš, případně další volitelná zařízení.
- Micro USB device konektor
- Ethernet (10/100T) konektor
- SD/multimediocard (MMC) konektor vložení karty je možno zvětšit paměťovou kapacitu desky.
- microSD/multimedacard (MMC) konektor a k desce se rovněž standardně dodává s kartou, na níž je nahrán software (deska defaultně běží pod Linuxem).
- SATA 7-pin data konektor
- Oneconsole UART
- VGA video output konektor
- Audio I/O jacky
- LVDS konektor – od firmy freescale je možné dokoupit displej 10 LVDS, jež je určen pro připojení na desku a zasazen v plastovém rámu.
- HDMI a LCD expansion port konektor

Jak je popsáno výše, deska obsahuje několik různých druhů výstupů pro video. Jsou zde výstupy pro HDMI, LCD a LVSD. Standardně je v desce přednastaveno napojení obrazu přes VGA. Následně je ale možné toto změnit, dle toho jaký druh displeje budeme používat.

Na desce je nainstalován operační systém LinuxUbuntu, ale je možné zvolit i jiný operační systém. Zde se nabízí například Windows EmbeddedCompact 7 nebo Android.



Obrázek 6.7: Vývojová deska Quick start board i.MX53

6.10 Implementace programu do zařízení

Aby bylo možné spouštět program na desce i.MX53 Quick Star Board je třeba jej na nejprve nahrát z počítače. Propojení s deskou je provedeno pomocí SSH protokolu. Samotný program a ostatní nutné části k jeho fungování jsou nahrávány na paměťovou kartu, jež byla za tímto účelem vložena do zařízení. Původně byla použita paměťová karta o velikosti 1GB, pro nahrání a kompilaci knihovny OpenCV však byla její kapacita nedostatečná. Proto je v současné době v desce paměťová karta o celkové kapacitě 4GB.

Prvním krokem ke spuštění programu na desce je nahrání zdrojových souborů knihovny openCV (verze 2.4.8) na paměťovou kartu. Velikost knihovny je zhruba 200MB. Během testování bylo zjištěno, že je potřeba knihovnu upravit pomocí `patchcap_libv4l.cpp` [18], aby bylo možné načítání obrazu z kamery.

Dalším krokem je kompilace knihovny, která byla provedena přímo v zařízení. Kvůli kompilaci na zařízení bylo nutné použít výše zmíněnou paměťovou kartu o celkové kapacitě 4GB. Během kompilace bylo potřeba paměti o kapacitě větší než je 1GB.

Následně bylo třeba nahrát na desku program. Aby nebylo nutné nahrávat knihovnu OpenCV a vytvořený program přímo na paměť desky, ale vše mohlo být uloženo na paměťové kartě, byl vytvořen symbolický link. Ten odkazuje na složku, jež obsahuje hlavičkové soubory uložené na kartě. Aby systém věděl, že sdílené knihovny nejsou uloženy jen ve složce „`/usr/lib`“

a „usr/local/lib“ byla upraven systémový soubor „etc/ld.so.conf“ o cestu ke sdíleným knihovnám na kartě a poté byl spuštěn příkaz „ldconfig“ aby systém dokázal sdílené knihovny za běhu najít. Další symbolický link byl vytvořen k jednoduššímu přístupu ke kartě. Respektive v domovském souboru uživatele (lucid) byl uložen pro dané paměťové zařízení název „karta“. Pak není nutné vždy zadávat původní název paměťového zařízení, který je dlouhý a nepraktický.

Dále, aby bylo možné v případě potřeby upravovat program přímo na desce, byl na desku nainstalován textový editor vim. Původně byl na desce textový editor vi, pro praktické využití je ale editor vim lepší.

Na kartě byla vytvořena složka „diplomka“ do níž jsou nahrávány všechny soubory požadované pro spuštění programu. V této složce se nachází samotný program (main.c), prvotní referenční obraz (reference.png) a soubor Makefile, který obsahuje příkazy pro kompilaci. Dále jsou na desce soubory, které jsou vytvořeny během kompilace.

Aby bylo možné program zkompilevat program, musí kompilátor znát přístup ke knihovně OpenCV, což je provedeno pomocí přepínače -L a cesty ke knihovně. Program je přeložen zadáním příkazu make a výsledný spustitelný soubor se jmenuje program. [5]

6.11 Testování

Testování programu v konečné podobě bylo provedeno reálně v automobilu, avšak na programu v počítači. Kamera byla přichycena nejprve na zpětné zrcátko, a to tak, aby byl v záběru jak spolujezdec, tak zadní sedadla automobilu. Testování probíhalo v několika měřeních. Při prvním testu se dvěma osobami na zadním sedadle a osobou na místě spolujezdce. Během tohoto testu však došlo k tomu, že osoba na předním sedadle částečně zakrývala osobu na sedadle za ní. Rovněž obličej osoby na sedadle spolujezdce zabíral příliš velký prostor obrazu a daným detekčním algoritmem nebylo možné bezpečně určit počet osob. Je však možné snímat oba prostory zvlášť a to sedadlo spolujezdce v jednom záběru a zadní sedadla ve druhém.

Na základě výše zmíněného byla kamera připevněna na strop automobilu tak, aby zabírala pouze osoby na zadním sedadle a zde bylo provedeno testování a to s různým počtem osob s různými osobami a na různých místech. Jednotlivé příklady rozmístění osob a postupu detekce je možné vidět na obrazcích 6.9. a 6.8. Naměřené výsledky je možné pozorovat v tabulce 1 a v tabulkách v příloze I. Tabulky obsahují především informace o počtu osob detekovaných na zadním sedadle. Dále pak informaci časovou hodnotou, kdy k detekci došlo. Detekce byla prováděna periodicky a byla zjištěna a uložena zhruba každé 2 s. U tabulek je také uvedeno datum, kdy k testování došlo a reálný počet osob přítomných při daném testování v záběru.

Tabulka 1 – test přesnosti detekce při přítomnosti 2 osob detekce, test proveden 26.4.2014

Počet detekovaných osob	Čas měření	Počet detekovaných osob	Čas měření
2	12:35:41	3	12:36:01
2	12:35:43	2	12:36:03
3	12:35:47	2	12:36:07
2	12:35:51	2	12:36:11
2	12:35:53	2	12:36:37
2	12:35:55	2	12:36:39
2	12:35:57	2	12:36:41

Celkem bylo provedeno 6 testovacích měření. Druhé měření bylo provedeno bez přítomnosti osob na zadním sedadle. Třetí s jednou osobou na zadním sedadle. Poté ještě opakovaně se dvěma a třemi osobami na zadním sedadle.

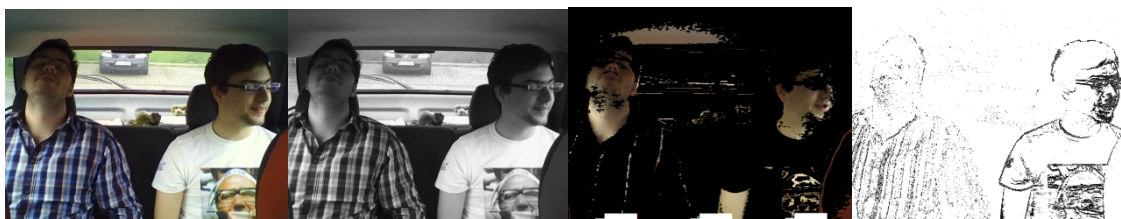
Měření vycházelo poměrně přesně, ačkoliv se během běhu programu objevovaly místy chyby, jak je možné vidět v tabulkách. Celkově na 70 měření připadá 7 chybně stanovené hodnot, pokud nebudeme započítávat případ s potiskem obličeje na tričku. Po převedení na procenta pak přesnost detekce daným algoritmem vychází na 90%. Při zahrnutí posledního testovaného měření s potiskem obličeje na tričku je hodnota podstatně nižší, protože téměř při všech měřeních byl zaznamenán jeden obličej navíc. Pak na 84 měření připadá 17 chyb, což sníží úspěšnost měření na 79,7%. Během měření byly osoby v automobilu jak plně oblečené, tak měli částečně odkryté ruce. Očekávání, že budou odkryté části detekovány, jako obličeje se nesplnilo. Během měření nebyly tyto části detekovány jako dostatečně velké oblasti dané barvy. Nicméně stále tato možnost zůstává jako problém, neboť potisk na tričku v dané barvě detekován byl. Prověření bylo provedeno i tak, že v následném měření byl potisk zakryt kusem látky. Pak proběhla detekce stejně, jako u ostatních testovacích měření. Výsledky tohoto měření jsou v tabulce č. 4.

Při testování nebyl nikdy v záběru řidič automobilu a to proto, že by se tím stěžovalo umístění kamery. Za předpokladu, že je auto v pohybu, musí být řidič vždy přítomen, proto není třeba jej detekovat pomocí kamery. Během testování rovněž nebyl zabírán spolujezdec. V prostředí ve kterém probíhalo testování a s dostupným vybavením nebylo možné kameru umístit tak, aby zabírala spolujezdce a zároveň zadní sedadla automobilu a bylo zajištěno, že nedojde k zákrytu osob.

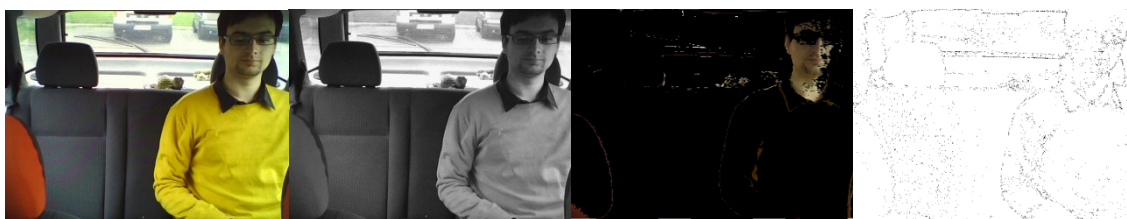
Co se týče chybových měření. Prakticky dochází při detekci k těmto nežádoucím jevům. Při detekci vznikají v obraze malé náhodné oblasti, které odpovídají barvou lidské kůži a zároveň pohybu. V testovaném automobilu je to například barva potahu (detekován také mírný

pohyb při jízdě automobilu). Pak může při výřezu podezřelé oblasti dojít k tomu, že je detekována pouze polovina obličeje a následně pak i obličej celý a informace je pak duplicitní. Tento problém byl během testování odhalen a na jeho základě byl změněn algoritmus tak, aby při detekci dostatečně velké oblasti, která by mohla obsahovat obličej, vymazal i data za hranicemi výřezové oblasti. Tím se zabrání tomu, aby byl jeden obličej detekován dvakrát. Nicméně díky tomu pak ve výřezových maticích mnohdy zůstane pouze část obličeje. Nicméně prakticky to detekci osoby nebrání.

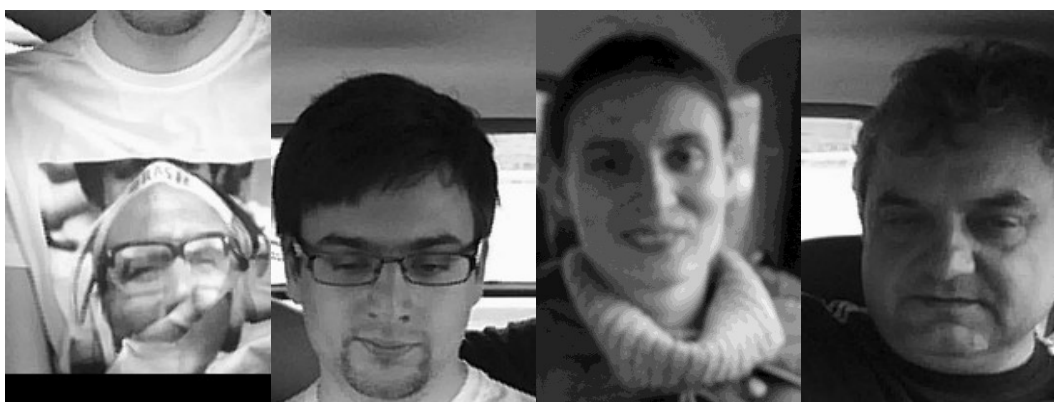
Dalším problém nastane, pokud se v obraze vyskytne oblast odpovídající svou barvou barvě pokožky a zároveň bude tato oblast dostatečně velká. K tomu může dojít odhalením větší části těla (stehna, ruce, hrud') nebo přítomností objektu, který rovněž barvou odpovídá (např. potisk obličeje na tričku. Proto by bylo dobré k programu přidat ještě další části, které by potvrdily, že hledaná oblast odpovídá obličeji člověka.



Obrázek 6.8: Příklad jednotlivých kroků detekce: zleva doprava: původní obraz, černobílý obraz, detekce barvy obličeje, detekce pohybu



Obrázek 6.9: Příklad jednotlivých kroků detekce: zleva doprava: původní obraz, černobílý obraz, detekce barvy obličeje, detekce pohybu



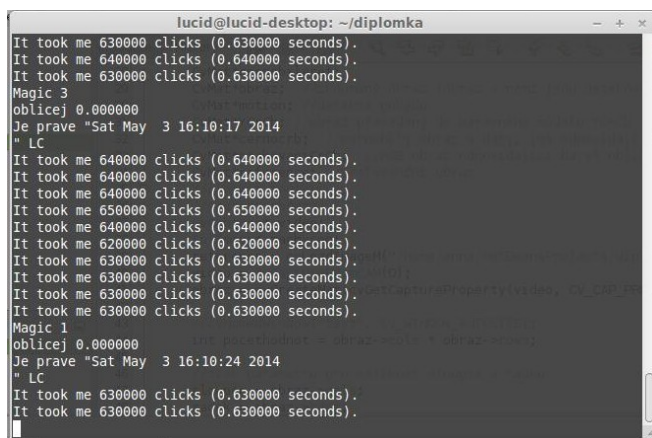
Obrázek 6.10: příklady detekovaných obličejů – výřezové obrazové matice

6.12 Časové zatížení programu

Důležitým požadavkem na navržený program je jeho časové zatížení. Rychlost s jakou dokáže zpracovat a vyhodnotit jeden snímek z kamery. Je třeba brát v potaz, že program běží opakovaně v nekonečné smyčce, respektive tak dlouho, jak trvá jízda automobilu. Během toho musí opakovaně a v krátkém časovém intervalu podávat informaci o počtu osob v automobilu tak, aby tato informace byla přesná a rychle dostupná. Během testování bylo měřeno časové zatížení programu a to na hlavní smyčce. Inicializace programu, připojení kamery a vytvoření většiny proměnných není do měření zatížení zahrnuto. Tato část programu však proběhne pouze jednou na začátku běhu. Dále se již neprovádí.

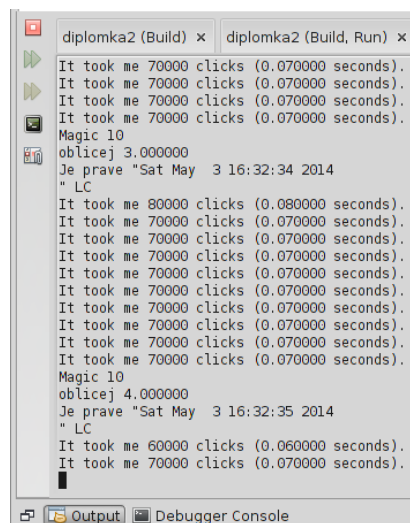
Testování proběhlo jak na stolním počítači, tak na vývojové desce i.MX53. Procesor na desce je výrazně pomalejší, než procesor v běžném stolním PC nebo notebooku. Což může být pro program značně omezující. Jak je vidět na obrázcích 6.11. a 6.12, na desce trvalo proběhnutí jednoho cyklu hlavní smyčky zhruba 620 – 650ms. V použitém notebooku (HP620) pak s rychlostí 60 – 80ms. Časové zatížení se může mírně lišit v závislosti na aktuálním snímku a na počtu detekovaných osob. Komplexnější snímky s větším počtem osob se budou obecně zpracovávat déle.

Na obrázcích 6.11. a 6.12 je také možné vidět zobrazení informace o počtu osob a informaci o čase.



```
lucid@lucid-desktop: ~/diplomka
It took me 630000 clicks (0.630000 seconds).
It took me 640000 clicks (0.640000 seconds).
It took me 630000 clicks (0.630000 seconds).
Magic 3
oblicej 0.000000
Je prave "Sat May 3 16:10:17 2014"
" LC
It took me 640000 clicks (0.640000 seconds).
It took me 640000 clicks (0.640000 seconds).
It took me 640000 clicks (0.640000 seconds).
It took me 650000 clicks (0.650000 seconds).
It took me 640000 clicks (0.640000 seconds).
It took me 620000 clicks (0.620000 seconds).
It took me 630000 clicks (0.630000 seconds).
It took me 630000 clicks (0.630000 seconds).
It took me 630000 clicks (0.630000 seconds).
It took me 630000 clicks (0.630000 seconds).
It took me 630000 clicks (0.630000 seconds).
Magic 1
oblicej 0.000000
Je prave "Sat May 3 16:10:24 2014"
" LC
It took me 630000 clicks (0.630000 seconds).
It took me 630000 clicks (0.630000 seconds).
```

Obrázek 6.11: Běh programu s měřením časového zatížení na desce. Spuštění přes terminál v PC



```
diplomka2 (Build) x diplomka2 (Build, Run) x
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
Magic 10
oblicej 3.000000
Je prave "Sat May 3 16:32:34 2014"
" LC
It took me 80000 clicks (0.080000 seconds).
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
It took me 70000 clicks (0.070000 seconds).
Magic 10
oblicej 4.000000
Je prave "Sat May 3 16:32:35 2014"
" LC
It took me 60000 clicks (0.060000 seconds).
It took me 70000 clicks (0.070000 seconds).
```

Obrázek 6.12: Běh programu s měřením časového zatížení v PC.

7 Závěr

Navrhnutý algoritmus pracuje na základě dvou možností detekce objektu a to na základě pohybu a barvy obličeje. Sama o sobě byla tato metoda vyhodnocena jako nedostačující. Za ideálních podmínek je poměrně přesná, ale existuje velké množství vnějších vlivů, díky kterým se může stát nepřesnou. Především oděni snímaných osob, neboť metoda je citlivá také na jiné plochy lidského těla, než je obličej. Především ruce a nohy. Tyto oblasti mohou být dostatečně velké, aby mohly být pokládány za obličej. Na zvážení je, zda se tento problém nedá eliminovat vyřazením oblastí, kde se obličej pravděpodobně nebude nacházet (například na podlaze automobilu). Tento přístup však prozatím nebyl testován.

Při testování bylo dosaženo poměrně dobrého výsledku, kdy byla přesnost detekce okolo 90% (79,8%). Většina chyb pak byly falešně pozitivní detekce. Tedy objekty, které nebyly obličej, byly jako obličej detekovány. Během testování nedocházelo k falešně negativním detekcím. Tedy počet detekovaných osob nebyl nižší, než byl počet osob skutečně v automobilu přítomných. Z praktického hlediska je pro danou aplikaci přípustnější falešně pozitivní detekce. Pokud bude nahlášeno více osob, dojde k menším škodám, než kdyby bylo nahlášených osob méně.

Původní idea pro detekci byla taková, že bude využito Houghovy transformace pro detekci kružnic a tedy i očí. Na základě detekce očí pak mělo být potvrzeno, že se jedná o obličej. Tato metoda však byla pro svou výpočetní náročnost z algoritmu vyřazena. Bylo by dobrou otestovat další metody pro detekce obličeje, jako je Haarova kaskáda, detekce textury obličeje, případně genetické algoritmy pro detekci kruhu. Je možné, že by tyto metody podávaly lepší výsledky.

Nicméně by nebylo dobré používat detekce na základě srovnání s šablonou, neboť tyto detekce vyžadují značné množství uložených šablon a také jsou považovány za pomalejší. Rovněž musí být šablony někde uloženy a v případě implementace na mikročip může být paměť zařízení nedostačující.

Dále během testování nastal problém se snímáním prostoru v automobilu, aby byly snímány všechny osoby najednou a nedocházelo k zákrytu osob. Tento problém by mělo být možné vyřešit buďto umístěním dvou kamer, z nichž by každá snímala jiný prostor, případně pomocí kamery s rybím okem.

I přesto, že dané navržená metoda detekce není citlivá na polohu obličeje, dá se předpokládat, že u některých případů bude tato detekční metoda selhávat. Příkladem můžou být osoby zahalené šátkem. Pokud nebude vidět dostatečný povrch kůže dané osoby, nebude ji možné nikterak detekovat. Ostatní algoritmy by si však s tímto problémem neporadily lépe. Z obecného hlediska, se nedá předpokládat, že algoritmus bude fungovat se 100% úspěšností.

Co se týče časových údajů, jsou k informaci o počtu osob přidávány automaticky. Na zvážení je, jak by zařízení v automobilu zjistilo, jaký je místní čas. Mohlo by dojít k nepřesnosti v případě přejetí z jednoho časového pásma do druhého. Zařízení, na kterých byl program

testován, a to jak v počítači, tak na desce, si byla schopna získat aktuální místní čas přes internet.

Celý navržený původní program má vyhovující časové zatížení a je jej možné využít jak na vývojové desce, tak v počítači a to v dostatečné rychlosti. V obou případech trvá zpracování jednoho snímku do 1s. Celkově je původní program v počítači zpomalován a výpis informací na obrazovku je realizován po 10 měřeních. Tento program je ve stejné podobě nahrán také na vývojové desce, kde však díky tomu pracuje výrazně pomaleji. Nicméně je možné toto upravit.

Celkově se nedá říci, že by metoda detekce osob v automobilu pomocí kamerového systému byla přesnější než detekce pomocí jiných metod. Sem patří detekce pomocí počtu zapnutých bezpečnostních pásů, případně pomocí tlakových senzorů v sedadle a případně v podlaze. Pravděpodobně nejlepší řešením by bylo využití všech těchto metod najednou. Aby to však přineslo požadovaný efekt, bude třeba dobře zvážit, jak dané metody správně zkombinovat, aby podávaly co nejlepší výsledky.

8 Zdroje

- [1] CHITRA, S.; BALKRISHNAN. *Comparative Study for Two Color Spaces HSCbCr and YCbCr in Skin Color Detection*. Applied Mathematical Sciences, Vol. 6, 2012, no. 85, 4229 - 4238
- [2] CASTLEMAN, Kenneth R. *Digital Image processing*. Upper Saddle River, New Jersey: Prenticehall, 1996. ISBN 0-13-211467-4.
- [3] KYUNGNAM, Kim. Face Recognition using Principle Component Analysis. Department of Computer Science University of Maryland, College Park, MD20742, USA
- [4] PHILIPS, Dwayne. [i]Image processing in C.[/i] Lawrence Kansas: R & D Publications, 1994. ISBN 0-13-104548-2. 2. digitalizované vydání v roce 2000. Dostupné také z: <http://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/cips2ed.pdf>
- [5] SOBELL, Mark G. [i]Mistrovství v Linuxu: příkazový řádek, shell, programování.[/i] 1. vyd. Brno: ComputerPress. 2007. ISBN 978-80-251-1726-2.
- [6] DOBEŠ, Michal. [i]Zpracování obrazu a algoritmy v C#.[/i] Praha : BEN - technická literatura, 2008. ISBN 978-80-7300-233-6.
- [7] JÁNOŠ, Jiří. Face detection in images. Brno. 2008, Bakalářská práce. Masarykova univerzita, Fakulta informatiky.
- [8] ČUNÁT, Petr. Začlenění letecké záchranné služby do systému eCall v České republice. Praha, 2012. Diplomová práce, České vysoké učení technické v Praze, Dopravní fakulta.
- [9] Heero, Harmonised eCall European Pilot. [online] citováno dne 9. 3. 2014. Dostupné na <http://www.heero-pilot.eu/view/cs/ecall.html>
- [10] CHUPEAU, Bertrand et all. Human face detection for automatic classification and annotation of personal photocollections. Francie. Research & Innovation, Corporate Recherche Rennes 1.
- [11] GARCIA, Christophe. TZIRITAS, Georgios. Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis. 1999. IEEE Transaction on multimedia, Vol 1. No.3.
- [12] HALOUNOVÁ, Lena. Zpracování obrazových dat. 2009, Praha, České vysoké učení technické v Praze. ISBN: 978-80-01-04253-3
- [13] OpenCV [online]. Citováno dne 1.3.2014. dostupné na <http://opencv.org/>
- [14] NOVOTNÝ, J. Auto přivolá při nehodě automaticky pomoc. *Právo*, 19.09.2013, p. 17.
- [15] ČSN EN 15722. *Inteligentní dopravní systémy - eSafety - Minimální soubor dat pro eCall*. Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2011. 30 p.

- [16] ČSN EN 16102. *Intelligentní dopravní systémy - eCall - Provozní požadavky na podporu eCall třetí stranou*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2012. 33 p.
- [17] ČSN EN 16072. *Intelligentní dopravní systémy - eSafety - Provozní požadavky na Panevropský eCall*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2012. 26 p.
- [18] BLAIR A. *cap_libv4l.cpp: Patch pro knihovnu OpenCV*. [online] staženo dne [12.4.2014], naposledy změněno [16.02.2014] dostupné na http://code.opencv.org/attachments/1547/cap_libv4l.cpp
- [19] OnStar. *SOS Emergency* [online], cit. dne [25.12.2013]. Dostupné z <https://www.onstar.com/web/portal/emergencyexplore?tab=1&g=1>
- [20] Toyota. *Safety connect*. Cit dne [25.12.2013]. dostupné na <http://www.toyota.com/safety-connect/#>
- [21] LANÍK, Ondřej. Citroën eTouch nyní v ČR: Zatím jen pro model C5 Exclusive [online]. *Auto*, 22. 06. 2010. cit dne [26.12.2013]. Dostupné na <http://www.auto.cz/c5-citroen-etouch-22521>
- [22] Ford. *911 Assist Overview* [online]. Cit dne [26.12.2013]. dostupné na <http://support.ford.com/sync-technology/911-assist-overview-sync>
- [23] BOSCH. *The Bosch eCall service ensures that life-saving assistance reaches the scene of an accident more quickly*. PressRelease. Srpen 2012.
- [24] Rowley, H., Baluja, S., Kanade, T., "Neural Network-Based Face Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23-38, 1998
- [25] online cit. dne [12.4.2014] dostupné na http://www.emeraldinsight.com/content_images/fig/2190220104031.png
- [26] VOLVO. *Volvo on call* [online] cit. dne [26.12.2014] dostupné na <http://www.volvocars.com/intl/sales-services/sales/volvo-on-call/Pages/default.aspx>
- [27] BMW. *BMW ConnectedDrive, BMW Assist* [online], cit dne [27.12.2014] dostupné na http://m.bmw.cz/m/mobi/cz/cs/general/bmw-connected-drive/bmw_assist.cml
- [28] Schuberth präsentiert erstes automatisches Notruf- und Ortungssystem für Motorradfahrer [online] cit dne [28.12.2014] naposledy upraveno dne [01.10.2012] dostupné na <http://www.schuberth.de/de/service/aktuell/news-detail/news/Premiere-RiderEcall-by-Schubert.html>

Příloha I – Výsledky testování

Tabulka 2 - test přesnosti detekce bez přítomnosti osob, test proveden 26.4.2014

Počet detekovaných osob	Čas měření	Počet detekovaných osob	Čas měření
0	12:40:27	0	12:45:12
0	12:40:29	0	12:45:15
0	12:40:32	0	12:45:17
0	12:40:34	1	12:45:20
0	12:40:36	0	12:45:23
0	12:40:39	0	12:45:25
0	12:40:41	0	12:45:27

Tabulka 3 – test přesnosti detekce při přítomnosti 1 osoby, test proveden 26.4.2014

Počet detekovaných osob	Čas měření	Počet detekovaných osob	Čas měření
1	12:57:52	1	12:58:09
1	12:57:54	1	12:58:11
1	12:57:56	1	12:58:13
1	12:57:59	1	12:58:20
1	12:58:01	1	12:58:22
1	12:58:03	1	12:58:24
1	12:58:06	1	12:58:27

Tabulka 4 – test přesnosti detekce při přítomnosti 3 osob, test proveden 27.4.2014

Počet detekovaných osob	Čas měření	Počet detekovaných osob	Čas měření
3	10:23:21	3	10:24:13
3	10:23:22	3	10:24:14
3	10:23:24	3	10:24:16
3	10:23:25	3	10:24:17

3	10:23:27	3	10:24:18
3	10:23:28	3	10:24:20
3	10:23:30	3	10:24:22

Tabulka 5 - test přesnosti detekce při přítomnosti 3 osob, jedna osoba měla oblečeno tričko s potiskem obličeje obrázek č. 11, test proveden 27.4.2014

Počet detekovaných osob	Čas měření	Počet detekovaných osob	Čas měření
3	10:23:21	3	10:24:13
4	10:23:22	4	10:24:14
4	10:23:24	4	10:24:16
3	10:23:25	4	10:24:17
4	10:23:27	3	10:24:18
4	10:23:28	4	10:24:20
4	10:23:30	4	10:24:22

Tabulka 6 – test přesnosti detekce při přítomnosti 2 osob, test proveden 27.4.2014

Počet detekovaných osob	Čas měření	Počet detekovaných osob	Čas měření
3	10:44:02	1	10:44:12
2	10:44:03	2	10:44:13
2	10:44:05	2	10:44:14
2	10:44:06	2	10:44:18
2	10:44:07	3	10:44:19
3	10:44:09	2	10:44:32
2	10:44:11	2	10:44:33

